VTU Syllabus

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 problems

CLO 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.

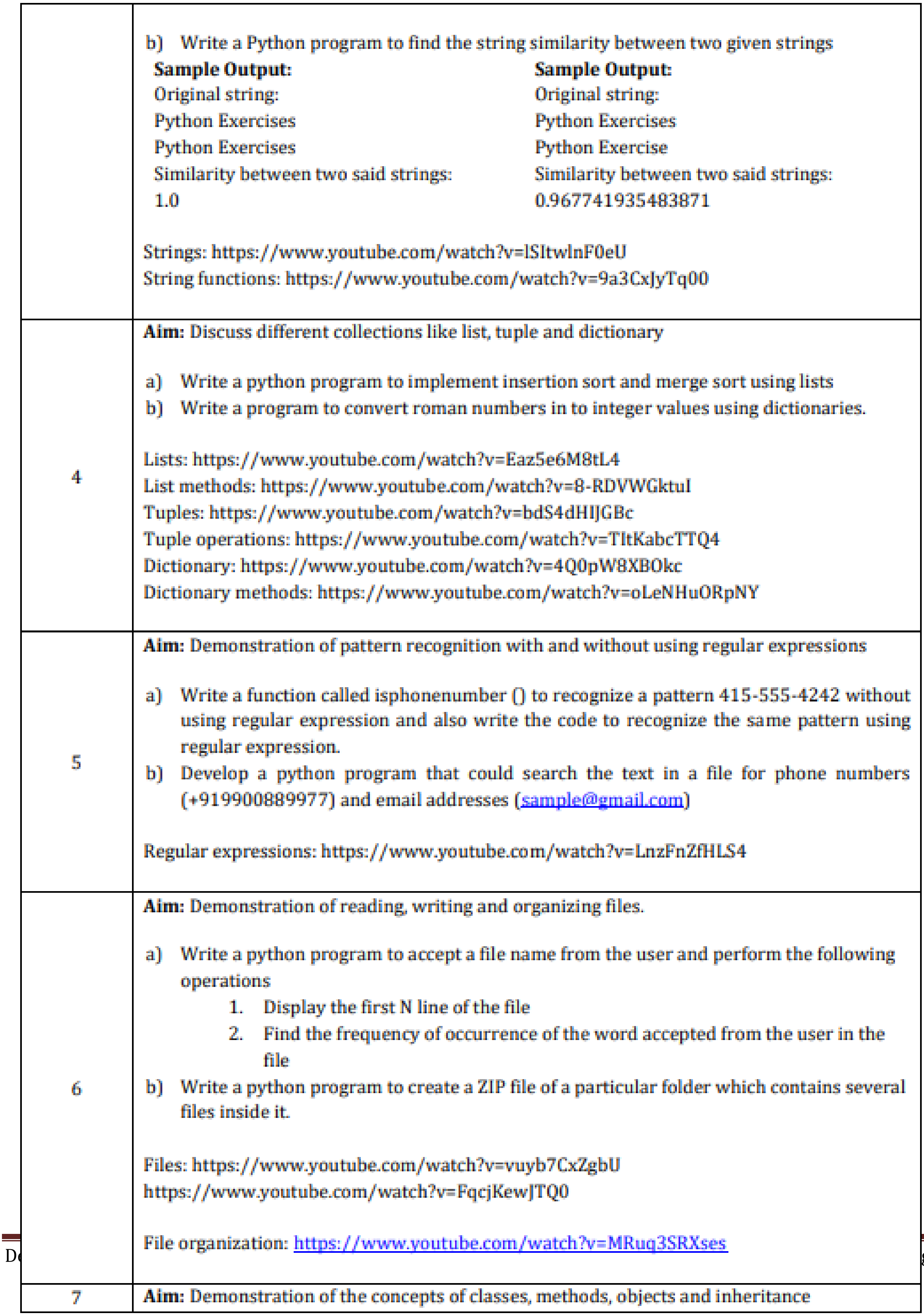
Prerequisite

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

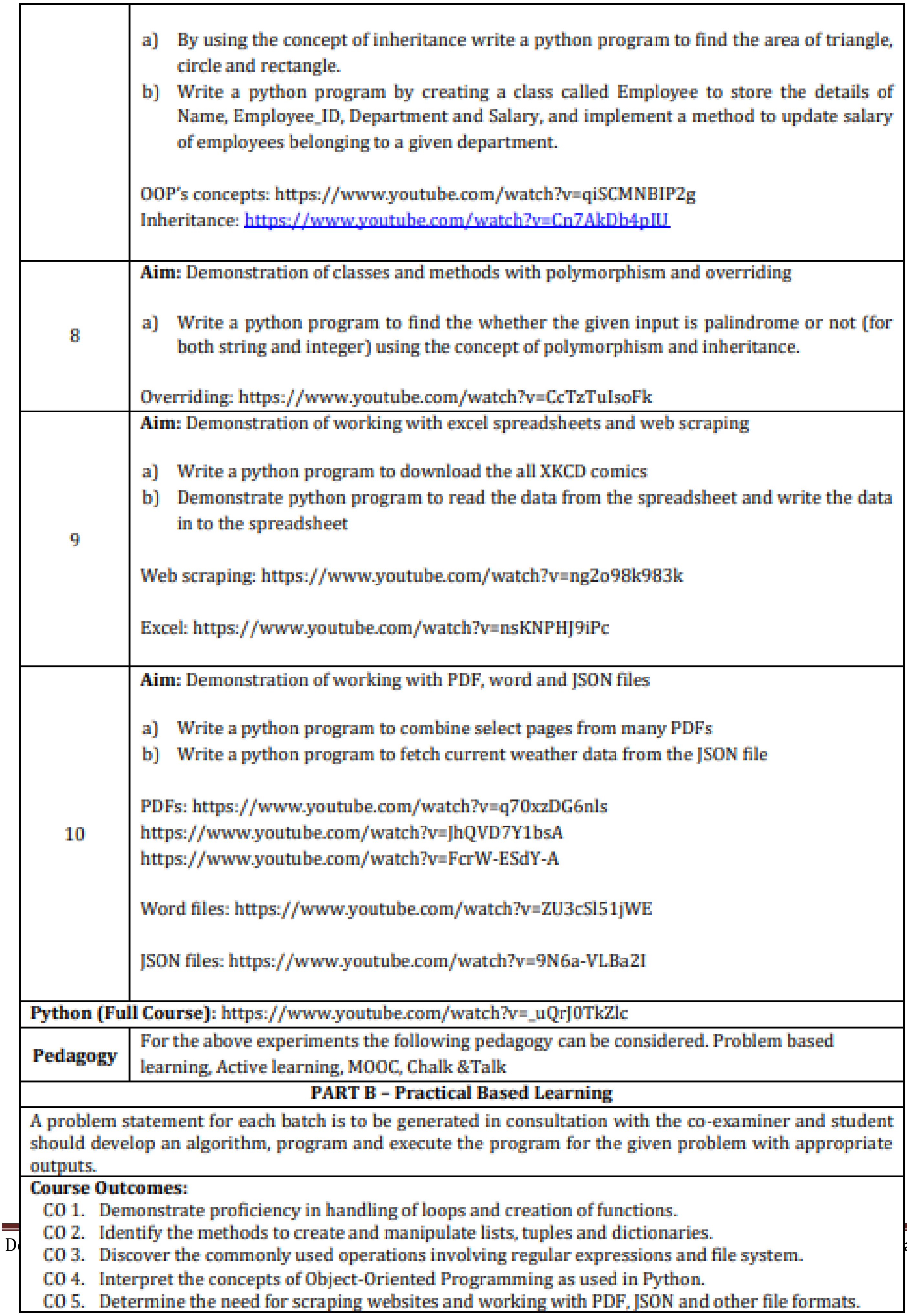
Python Installation: https://www.youtube.com/watch?v=Kn1HF3oD19c

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

Sl. No.	***PART A – List of problems for which student should develop program and execute in the Laboratory***
1	**Aim:** Introduce the Python fundamentals, data types, operators, flow control and exception handling in Python a) Write a python program to find the best of two test average marks out of three test's marks accepted from the user. 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. Datatypes: https://www.youtube.com/watch?v=gCCVsvgR2KU Operators: https://www.youtube.com/watch?v=v5MR5JnKcZI Flow Control: https://www.youtube.com/watch?v=PqFKRqpHrjw For 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: 1.0 **Sample Output:** Original string: Python Exercises Python Exercise Similarity between two said strings: 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-RDVWGktuI Tuples: 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 the following operations 1. Display the first N line of the file 2. Find the frequency of occurrence of the word accepted from the user in the file b) Write a python program to create a ZIP file of a particular folder which contains several 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



<table>
<tr><td></td><td>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</td></tr>
<tr><td>8</td><td><b>Aim:</b> 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 (for both string and integer) using the concept of polymorphism and inheritance.

Overriding: https://www.youtube.com/watch?v=CcTzTuIsoFk</td></tr>
<tr><td>9</td><td><b>Aim:</b> 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

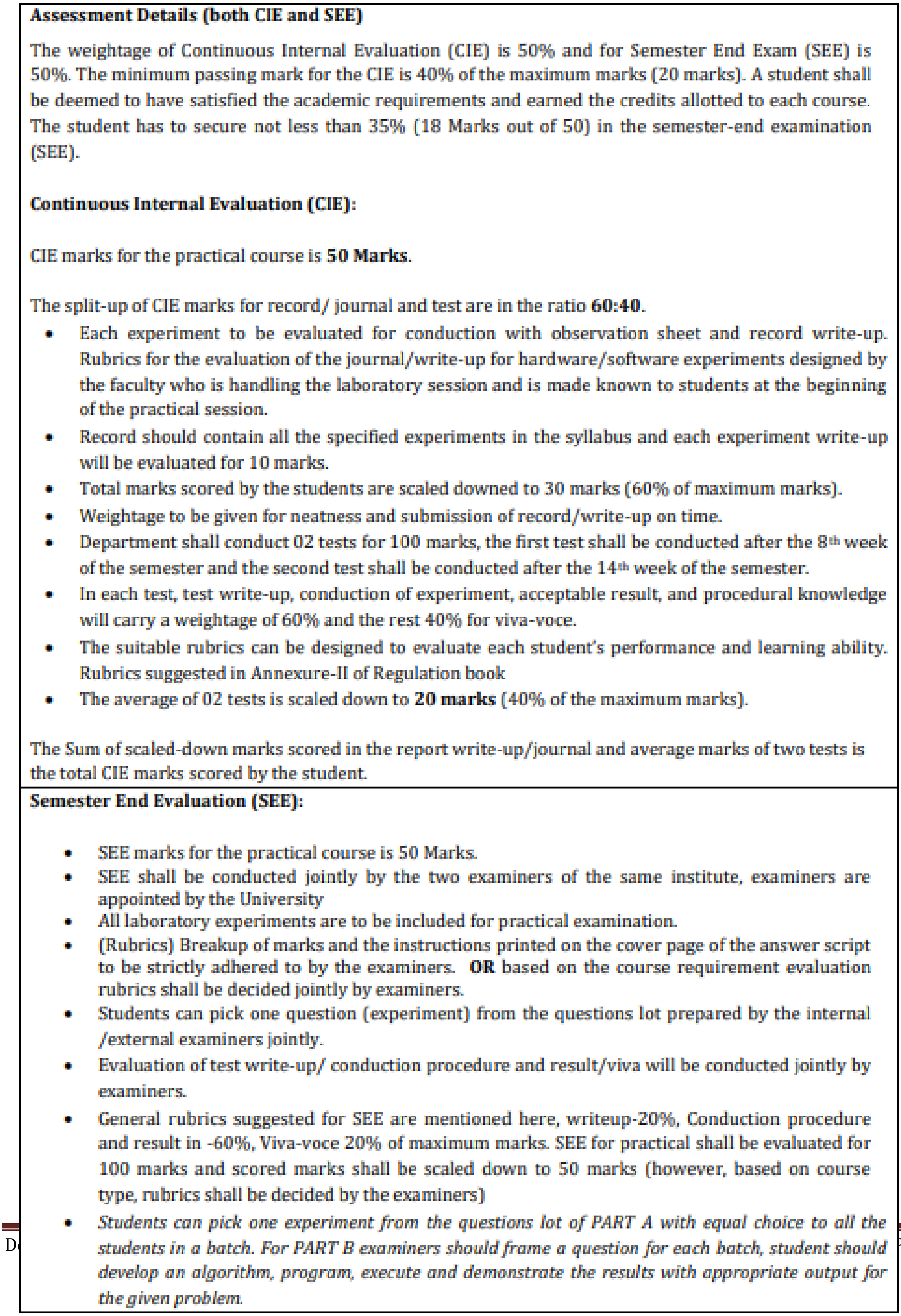
Excel: https://www.youtube.com/watch?v=nsKNPHJ9iPc</td></tr>
<tr><td>10</td><td><b>Aim:</b> Demonstration of working with PDF, word and JSON files

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</td></tr>
<tr><td colspan="2"><b>Python (Full Course):</b> https://www.youtube.com/watch?v=_uQrJ0TkZlc</td></tr>
<tr><td><b>Pedagogy</b></td><td>For the above experiments the following pedagogy can be considered. Problem based learning, Active learning, MOOC, Chalk &Talk</td></tr>
<tr><td colspan="2"><b>PART B – Practical Based Learning</b></td></tr>
<tr><td colspan="2">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.</td></tr>
<tr><td colspan="2"><b>Course Outcomes:</b>
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.</td></tr>
</table>



Assessment 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).

Continuous Internal Evaluation (CIE):

CIE marks for the practical course is **50 Marks**.

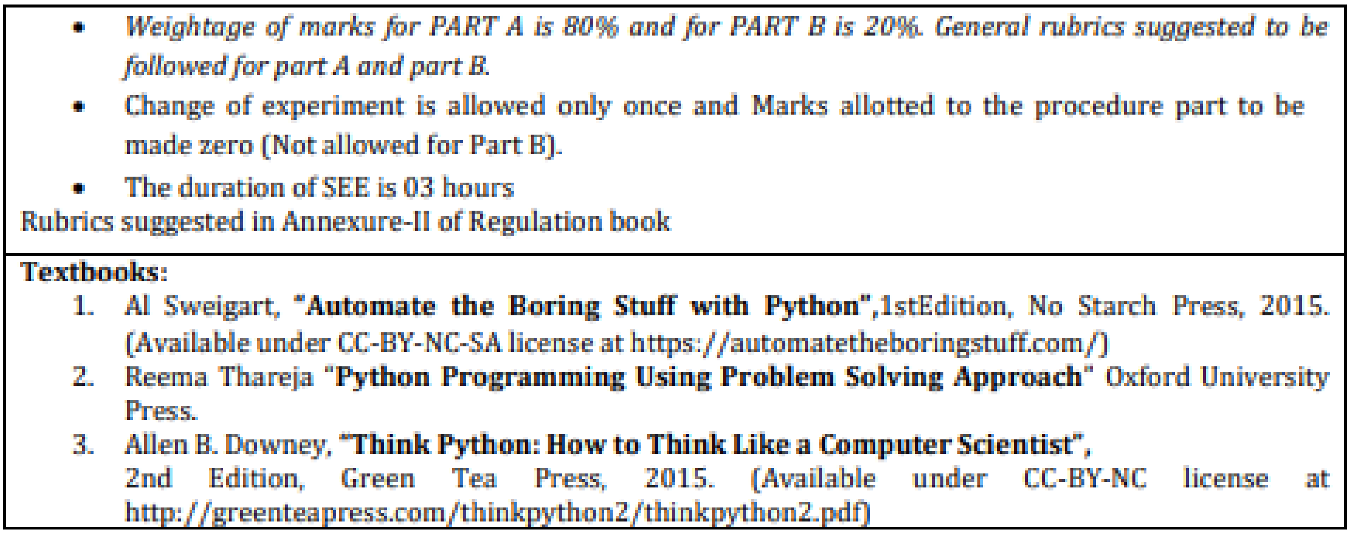
The split-up of CIE marks for record/ journal and test are in the ratio **60:40**.

- 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.
- The suitable rubrics can be designed to evaluate each student's performance and learning ability. 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.
- Evaluation of test write-up/ conduction procedure and result/viva will be conducted jointly by examiners.
- General rubrics suggested for SEE are mentioned here, writeup-20%, Conduction procedure and result in -60%, Viva-voce 20% of maximum marks. SEE for practical shall be evaluated for 100 marks and scored marks shall be scaled down to 50 marks (however, based on course type, rubrics shall be decided by the examiners)
- *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 for the given problem.*



- *Weightage of marks for PART A is 80% and for PART B is 20%. General rubrics suggested to be followed for part A and part B.*
- Change of experiment is allowed only once and Marks allotted to the procedure part to be made zero (Not allowed for Part B).
- The duration of SEE is 03 hours

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

Aim 1: 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.

```
tm1 = int(input("Enter First Test Marks"))
tm2 = int(input("Enter Second Test Marks"))
tm3 = int(input("Enter Third Test Marks"))

if tm1>tm2 and tm1>tm3 :                   # is tm1 is largest
    if tm2>tm3:                            # is tm2 is second largest
        avg = (tm1+tm2)/2                  # compute average of tm1 and tm2
    else:
        avg = (tm1+tm3)/2                  # else compute average of tm1 and tm3
elif tm2>tm1 and tm2>tm3:                   # is tm2 is largest
    if tm1>tm3:                            # is tm1 is second largest
        avg = (tm1+tm2)/2                  # compute average of tm2 and tm1
    else:
        avg = (tm2+tm3)/2                  # else compute average of tm2 and tm3
else:                                      # Else tm3 is the largest
    if tm1>tm2:                            # is tm1 is second largest
        avg = (tm3+tm1)/2                  # compute average of tm3 and tm1
    else:
        avg = (tm3+tm2)/2                  # else compute average of tm3 and tm2

print("The average of two best test marks out of three test marks = ",avg)
```

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.

```
x = int(input("Enter a number: "))
c0,c1,c2,c3,c4,c5,c6,c7,c8,c9=0,0,0,0,0,0,0,0,0,0
num = x
rev = 0
while x>0:
    r = x%10
    rev = rev*10 + r
    x = x//10

    if r == 0:
        c0+=1
    elif r == 1:
        c1+=1
    elif r == 2:
        c2+=1
    elif r == 3:
        c3+=1
    elif r == 4:
        c4+=1
    elif r == 5:
        c5+=1
    elif r == 6:
        c6+=1
    elif r == 7:
        c7+=1
    elif r == 8:
        c8+=1
    elif r == 9:
        c9+=1

if rev==num:
    print("The Number {0} is palindrome ".format(num))
else:
    print("The Number {0} is not palindrome ".format(num))

print("The occurrence of 0 is {0}, 1 is {1}, 2 is {2}, 3 is {3}, 4 is {4}, 5 is {5}, 6 is {6}, 7 is {7}, 8 is {8}, 9 is
                                                    {9}".format(c0,c1,c2,c3,c4,c5,c6,c7,c8,c9))
```

Aim 2: 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 Fibo(n):                                      # function F definition
    ft=0                                          # initialize first term to 0
    st=1                                          # initialize second term to 1
    tt=0                                          # initialize third term to 0
    if n <= 0:                                    # check if n is greater than 0
        print ("The value of N is expected to be greater than 0 but received ",n)
    else:
        print("The ",n," terms of the Fibonacci are ")
        print(ft,'\n',st)                         # print the default terms 0 and 1 of the Fibonacci series
        n=n-2                                     # decrement n by 2 as first and second terms are generated already
        while (n>0):                              # while n is greater than 0 do the following
            tt = ft + st                          # third term is sum of first term and second term
            print(tt)                             # print third term
            ft = st                               # assign second term to first term
            st = tt                               # assign third term to second term
            n-=1                                  # decrement the value of n by 1

n = int(input("Enter the value of n: "))          # read the number of terms of the Fibonacci series
Fibo(n)                                           # invoke the function F to generate the Fibonacci series of n terms
```

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

```
def BinToDec(x):                    # Definition of Binary To Decimal Conversion function
    dec = 0
    i = 0
    while x>0:                      # till x > 0 do the followings
        r = x%10                    # get the least significant bit of the Binary number
        if r !=0 and r !=1:         # if its is not a binary digit
            print("Enter a valid Binary number")
            return 0                # then return 0
        else:
            dec = dec + r*2**i      # else compute the decimal equivalent value i.e. bit x 2^position
            x = x // 10             # get the remaining bits of the binary number
            i += 1                  # increment i by one i.e get the next significant bit and repeat
    return dec                      # if x==0,then return the decimal equivalent value of Binaray number x

def OctaToHexa(n):                  # Definition of Octal To Hexa Decimal Conversion function
    num = n
    dec = 0
    base = 1
    temp = num                      # assign number to temp
    while temp:                     # till temp value is True (!0) do the following
        r = temp % 10               # take the Least significant digit of the Octal number
        temp = temp // 10           # update temp, value by removing the least significant digit of the Octal no
        dec += r * base             # compute the Decimal value of the least significant digit of the Octal no.
        base = base * 8             # update the positional value of the next digit and repeat

    result = ''                     # initialize result to 0

    while dec != 0:                 # till decimal equivalent value of the Ocatl nu (dec) is >0 do the following
        temp = 0
        temp = dec % 16             # get Hex value of least significant digit of Decimal number
        if temp < 10:               # if the Hex value is single digit number
            result = str(temp) + result   # convert it into string and accumulate in Result
        else:
            result = chr(temp + 87) + result # else if the Hex value is more than 9 then get the equivalent Hex digit
        dec = dec // 16             # get the next Hex value and repeat
    return result                   # return the Hex Decimal equivalent of the Decimal number

x = int(input("Enter a Binary number "))
result = BinToDec(x)
if result:
    print("The Decimal equivalent of {0} is {1}".format(x, result))


y = int(input("Enter a Octal number "))
result = OctaToHexa(y)
```

```
if result:
    print("The Hexa Decimal equivalent of {0} is {1}".format(y, result))
```

Aim 3: Demonstration of manipulation of strings using string methods

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

```
x = input("Enter a sentence")                  # Read a string
y = x
print("There are",len(x.split())," words in the sentence")  # split the string into words and compute the length
digits,upper,lower=0,0,0
for i in x:                                    # For each character i in the read string x
    if i.isdigit():                            # if the character i is a digit increment the Digit count
        digits+=1
    elif i.isupper():                          # if character i is a upper case character increment upper count
        upper+=1
    elif i.islower():                          # if character i is a lower case character increment lower count
        lower+=1
print("There are {0} digits, {1} upper case characters and {2} lower case characters in the sentence".format(digits,upper,lower))
```

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

```
if __name__== '__main__':
    x = input("Enter first String")            # Read the first string
    y = input("Enter second String")           # Read the second string

    x = x.strip()                   # Strip extra spaces
    y = y.strip()
    sim=0                           # Initialize the similarity count to 0

    if len(x)>len(y):               # Compute the longer string
        xx = x                      # Assign it to xx
        yy = y                      # yy is the shorter string
    else:
        xx = y
        yy = x

    j=0

    for i in yy:                    # For each character in yy (the shorter string)
        if i==xx[j]:                # check whether it is equal to the character in xx (the longer string)
            sim+=1                  # if so increment the similarity count by 1
        else:
            pass                    # else do no do anything
        j+=1                        # increment the index of longer string
```

```
similarity = (sim/len(xx))                # compute similarity = (similarity between strings) / length of longer string

print("The similarity between the two given strings is", similarity)
```

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

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

```
def InsertionSort(lst):
    i=0
    while i<len(lst):                       #Till i less than the length of the list repeat
        small = lst[i]
        for j in range(i+1,len(lst)):       # Find the smallest element of the list
            nxt =lst[j]
            if small>nxt:
                small = lst[j]
        index = lst.index(small)            # index is the index of the smallest element
        if i==index :                       # if element is at its correct position then do nothing
            pass
        else:
            lst.remove(small)               # Else move the smallest to its correct position
            lst.insert(i,small)
        i=i+1
    return lst

def mergeSort(arr):
    if len(arr) > 1:
        mid = len(arr)//2                   # Create sub_array2 ← A[start..mid]
        sub_array1 = arr[:mid]              # and sub_array2 ← A[mid+1..end]
        sub_array2 = arr[mid:]

        mergeSort(sub_array1)               # Sort the two halves
        mergeSort(sub_array2)

        i = j = k = 0                       # Initial values for pointers to keep track of where we are in each array

       while i < len(sub_array1) and j < len(sub_array2):
            if sub_array1[i] < sub_array2[j]: # Until we reach the end of either start or end, pick larger among
                 arr[k] = sub_array1[i]      # start & end & place them in the correct position in sorted array
                 i += 1
            else:
                 arr[k] = sub_array2[j]
                 j += 1
            k += 1

       while i < len(sub_array1):            # When all elements are traversed in either arr1 or arr2,
            arr[k] = sub_array1[i]          # pick up the remaining elements and put in sorted array
            i += 1
            k += 1
      while j < len(sub_array2):
            arr[k]  =  sub_array2[j]
            j += 1
            k += 1
    return arr
```

```
if__name__=='__main__':
    lst = []
    n = int(input("Enter the size of the list"))
    print("Enter ",n," numbers of the list")
    for i in range(0,n+1):
        lst.append(int(input()))
    print(lst)

    print("Enter 1: Insertion Sort, 2: Merge Sort ")
    ch = int(input())
    if ch==1:
        lst = InsertionSort(lst)
        print("The sorted array is ",lst)
    elif ch==2:
        lst = mergeSort(lst)
        print("The sorted array is ",lst)
    else:
        print("Invalid Choise")
```

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

```
def roman_to_int(s):
    rom_val = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
    int_val = 0
    for i in range(len(s)):
        if i > 0 and rom_val[s[i]] > rom_val[s[i - 1]]:
            int_val += rom_val[s[i]] - 2 * rom_val[s[i - 1]]
        else:
            int_val += rom_val[s[i]]
    return int_val

x = input(“Enter the Roman Number”)
print(roman_to_int(x.upper())
```

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

5 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.

```
def isphonenumber(x):
    l = len(x)
    if l!=12:
        return 0
    else:
        for i in range(0,l):
            if i==0 or i==1 or i==2 :
                if x[i].isdigit() == False:
                    return 0
            if i==4 or i==5 or i==6 :
                if x[i].isdigit() == False:
                    return 0
            if i==8 or i==9 or i==10 :
                if x[i].isdigit() == False:
                    return 0
            if i==3 or i==7:
                if x[i] != '-':
                    return 0
        return 1

def REisphonenumber(x):
    import re
    pno = re.compile(r'\d\d\d-\d\d\d-\d\d\d\d')
    mo = pno.search(x)
    if mo == None:
        return 0
    else:
        return mo.group()

if__name__=='__main__':
    phoneNo = input("Enter a phone number of ddd-ddd-dddd format to validate")
    ch = int(input("Enter 1.To validate without RE. 2. To validate using RE"))
    if ch==1:
        if isphonenumber(phoneNo)==1:
            print("You have entered a valid phone number")
        else:
            print("You have entered an invalid phone number")
    elif ch==2:
        if REisphonenumber(phoneNo)==1:
            print("You have entered a valid phone number")
        else:
            print("You have entered an  invalid phone number")
```

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

```
import re

try:
    file = open("data.txt")
    for line in file:
        line = line.strip()
        match = re.findall(r"(\d{10})", line)
        if(len(match)>0):
            print(match)
        emails = re.findall("[0-9a-zA-z]+@[0-9a-zA-z]+\.[0-9a-zA-z]+", line)
        if(len(emails) > 0):
            print(emails)
except FileNotFoundError as e:
    print(e)
```

Aim 6: Demonstration of reading, writing and organizing files

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

1. Display the first N line of the file

2. Find the frequency of occurrence of the word accepted from the user in the file

```
inputFile = "abc.txt"                           # input text file

N = int(input("Enter N value: "))               # Enter N value

with open(inputFile, 'r') as filedata:          # Opening the given file in read-only mode
 linesList= filedata.readlines()                # Read the file lines using readlines()
 print("The following are the first",N,"lines of a text file:")

   for textline in (linesList[:N]):             # Traverse in the list of lines to retrieve the first N lines of a file
     print(textline, end ='')                   # Printing the first N lines of the file line by line.

word=input("Enter word to be searched:")
k = 0

with open(inputFile, 'r') as f:
   for line in f:
     words = line.split()
     for i in words:
       if(i==word):
         k=k+1
print("Occurrences of the word:")
print(k)

filedata.close()                                # Closing the input file
```

6 b) Write a python program to create a ZIP file of a particular folder containing several files inside it.

```
from zipfile import ZipFile
import os
from os.path import basename

# create a ZipFile object
with ZipFile('sampleDir.zip', 'w') as zipObj:
  # Iterate over all the files in directory
  for folderName, subfolders, filenames in os.walk("ITLAB10"):
    for filename in filenames:
      #create complete filepath of file in directory
      filePath = os.path.join(folderName, filename)
      # Add file to zip
```

```
            zipObj.write(filePath, basename(filePath))
```

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

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

```
class Shape:
    area,radius,length,breadth,a,b,c=0,0,0,0,0,0,0
    def___init_(self, r):
        self.radius = r
    def___init_(self, l,b):
        self.length = l
        self.breadth = b
    def___init_(self, a,b,c):
        self.a = a
        self.b = b
        self.c = c

class Circle(Shape):
    def___init_(self,r):
        super().__init__(r,0,0)

    def area(self,r):
        print("The area of circle is ",3.14*r*r)

class Rectangle(Shape):
    def___init_(self,l,b):
        super().__init__(0,l,b)

    def area(self,l,b):
        print("The area of rectangle is ",l*b)

class  Triangle(Shape):
    def___init_(self,a,b,c):
        super().__init__(a,b,c)

    def area(self,a,b,c):
        s = (a+b+c)/2
        import math
        print("The area of triangle is ",math.sqrt(s*(s-a)*(s-b)*(s-c)))



if___name__=='__main__':
    print("Enter your choice to compute area of 1.Circle, 2.Rectangle, 3.Triangle")
    ch = int(input())
    if ch==1:
        r = int(input("Enter radius of circle"))
        c = Circle(r)
        c.area(r)
    elif ch==2:
        l = int(input("Enter length of rectangle"))
        b = int(input("Enter breadth of rectangle"))
        r = Rectangle(l,b)
        r.area(l,b)
    elif ch==3:
        a = int(input("Enter length of side a of triangle"))
        b = int(input("Enter length of side b of triangle"))
        c = int(input("Enter length of side c of triangle"))
        t = Triangle(5,6,7)
        t.area(5,6,7)
    else:
        print("Invalid choice")
```

7 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.

```
class Employee:
    def___init_(self,name,eid,dept,sal):
        self.name = name
        self.eid = eid
        self.dept = dept
        self.sal = sal
    def salUpdate(self,eid,dept,updsal):
        self.sal =  updsal

if___name___== "_main_":
    emp=[]
    while True:
        ch = int(input("\n Enter 1.Create Employee\n 2.To display all employees\n 3.To Update an employee
                        salary\n 4.To exit\n"))
        if ch==1:
            n = input("Employee Name: ")
            i = int(input("Employee ID: "))
            d = input("Employee Department: ")
            s = int(input("Employee Salary: "))
            emp.append(Employee(n,i,d,s))
            print("Employee details created",len(emp))
        elif ch==2:
            for i in emp:
                print("\n Employee Name:{0}\n Employee ID:{1}\n Employee Department:{2}\n Employee
                                      Salary:{3}".format(i.name, i.eid, i.dept, i.sal) )
        elif ch==3:
            upd=0
            print("\n Enter the Department and ID of the employee to update salary")
            empid = int(input("Employee ID: "))
            dept = input("Employee Department: ")
            for i in emp:
                if empid == i.eid and dept == i.dept:
                    upsal = int(input("\n Enter the updated salary"))
                    i.salUpdate(empid,dept,upsal)
                    print("\n Salary updated")
                    upd=1
            if upd==0:
                print("\n Employee does not exist")
        elif ch==4:
            break;
        else:
            print("Invalid choice")
```

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

8. 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 strPalindrome:
    def___init_(self):
        self.word=""
        self.ll=""
    def check(self,s):
        self.word = list(s)
        ll=self.word.copy()
        self.word.reverse()
        if (ll==self.word):
            print("\nIt is Palindrome")
        else:
            print("\nIt is Not Palindrome")
class noPalindrome(strPalindrome):
    def___init_(self):
        super().__init___
    def check(self,no):
        super().check(str(no))


if___name_=='_main_':
    while True:
        ch = int(input("Enter 1.For String palindrome 2.For Integer Palindrome 3.To Exit : "))
        if ch==1:
            text = input("Enter a string to check : ")
            s = strPalindrome()
            s.check(text)
        elif ch==2:
            text = int(input("Enter a number to check : "))
            s = noPalindrome()
            s.check(text)
        elif ch==3:
            break
        else:
            print("Invalid choice")
```

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

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

```
import requests, os, bs4
url = 'http://xkcd.com'                          # starting url
os.makedirs('xkcd',exist_ok = True)              # create a directory to store all the downloads

while not url.endswith("#"):
                                                 # Download the page.
        print("Downloading the page ... ")
        res = requests.get(url)
        res.raise_for_status()
        try:
                soup = bs4.BeautifulSoup(res.text,'lxml')
        except bs4.FeatureNotFound:  # lxml is not installed
                soup = bs4.BeautifulSoup(res.text,'html.parser')

                                                 # Find the URL of the comic image.
        comic_element = soup.select('#comic img')
        if comic_element == []:
                print("No comic image found!!..")
        else:
                comic_image_url = comic_element[0].get('src')
                                                 # download the image
                print("Downloading the image %s .. " %(comic_image_url))
                res = requests.get('http:' + comic_image_url)
                res.raise_for_status()

                                                 # Save the image to ./xkcd.
                file = open( os.path.join('xkcd',os.path.basename(comic_image_url)) , 'wb')
                for chunk in res.iter_content(10000):
                        file.write(chunk)
                file.close()

                                                 # Get the Prev button's url.
        prev_link = soup.select('a[rel="prev"]')[0]
        url = 'http://xkcd.com' + prev_link.get('href')

print("Done")
```

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

```
import openpyxl                                   # import openpyxl module
path = "abc.xlsx"                                 # Give the location of the file
wb_obj = openpyxl.load_workbook(path)             # To open the workbook workbook object is created
sheet_obj = wb_obj.active                         # Get workbook active sheet object from the active attribute
row = sheet_obj.max_row                           # Getting the value of maximum rows and column
column = sheet_obj.max_column

print("Total Rows:", row)
print("Total Columns:", column)

print("\nValue of first column") # printing the value of first column Loop will print all values of first column
for i in range(1, row + 1):
    cell_obj = sheet_obj.cell(row = i, column = 1)
    print(cell_obj.value)

print("\nValue of first row")             # printing the value of first column Loop will print all values of first row
for i in range(1, column + 1):
    cell_obj = sheet_obj.cell(row = 2, column = i)
    print(cell_obj.value, end = " ")


import openpyxl                           # import openpyxl module

# import openpyxl module # Call a Workbook() function of openpyxl to create a new blank Workbook object
wb = openpyxl.Workbook()

sheet = wb.active                         # Get workbook active sheet from the active attribute

# Cell objects also have row, column and coordinate attributes that provide location information for the cell.
# Note: The first row or column integer is 1, not 0. Cell object is created by using sheet object's cell() method.
c1 = sheet.cell(row = 1, column = 1)

c1.value = "Hello"                        # writing values to cells
c2 = sheet.cell(row= 1 , column = 2)
c2.value = "World"

# Once have a Worksheet object, one can access a cell object by its name. A2 means column = 1 & row = 2.
c3 = sheet['A2']
c3.value = "Welcome"

c4 = sheet['B2']                          # B2 means column = 2 & row = 2.
c4.value = "Everyone"

# Anytime you modify the Workbook object or its sheets and cells, the spreadsheet file will not be saved until
```

```
#you call the save() workbook method.
wb.save("abc.xlsx")
```

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

10 a) Write a python program to combine select pages from many PDFs

```
from PyPDF2 import PdfFileMerger

#Create an instance of PdfFileMerger() class
merger = PdfFileMerger()

#Create a list with the file paths
pdf_files = ['sample_page1.pdf', 'pdf_files/sample_page2.pdf']

#Iterate over the list of the file paths
for pdf_file in pdf_files:
   #Append PDF files
   merger.append(pdf_file)

#Write out the merged PDF file
merger.write("merged_2_pages.pdf")
merger.close()
```

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

```
import requests, json                                          # importing requests and json
BASE_URL = https://api.openweathermap.org/data/2.5/weather?    # base URL
CITY = "Hyderabad"
# API key API_KEY = "Your API Key"
URL = BASE_URL + "q=" + CITY + "&appid=" + API_KEY             # upadting the URL
response = requests.get(URL)                                   # HTTP request
if response.status_code == 200:                          # checking the status code of the request
    data = response.json()                                     # getting data in the json format
    main = data['main']                                        # getting the main dict block
    temperature = main['temp']                                 # getting temperature
    humidity = main['humidity']                                # getting the humidity
    pressure = main['pressure']                                # getting the pressure
    report = data['weather']                                   # weather report
    print(f"{CITY:-^30}")
    print(f"Temperature: {temperature}")
    print(f"Humidity: {humidity}")
    print(f"Pressure: {pressure}")
    print(f"Weather Report: {report[0]['description']}")
else:
    print("Error in the HTTP request")                         # showing the error message
```