

SUBJECT: PYTHON PROGRAMMING LABORATORY

SUBJECT CODE: 21CSL46

IV SEMESTER



PREPARED BY:
MONISHA R
ASSISTANT PROFESSOR

Program-01 A

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.

```
marks1=int(input("Enter Test 1 Marks:"))  
marks2=int(input("Enter Test 2 Marks:"))  
marks3=int(input("Enter Test 3 Marks:"))  
minimum=min(marks1,marks2,marks3)  
sumofbest2=marks1+marks2+marks3-minimum  
avgofbest2=sumofbest2/2  
print("Average of best Two :",avgofbest2)
```

OUTPUT:

Program-01 B

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.

```
val = int(input("Enter a value : "))
str_val = str(val)
if str_val == str_val[::-1]:
    print("Entered Value is Palindrome")
else:
    print("Entered Value is Not a Palindrome")
for i in range(10):
    if str_val.count(str(i)) > 0:
        print(str(i), "appears", str_val.count(str(i)), "times")
```

OUTPUT:

Program-02 A

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

a) Defined as a function F as $F_n = F_{n-1} + F_{n-2}$. 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.

#define the Function
def fn(n):

#if Entered Number is Equal To 1 Returns the 0
if n == 1:

return 0

#else If the Entered Number is Equal to 2 Returns the 1

elif n == 2:

return 1

else:

#else return function number

return fn(n - 1) + fn(n - 2)

#enter the Integer Value

num = int(input("Enter a Number:"))

#If Entered Value is Greater the 0

if num > 0:

#Display the enterd number and fibenacci number

print("fn(", num, ") = ", fn(num), sep="")

else:

#display the If entered number is less than 0

print("Error in input")

OUTPUT:

Program-02 B

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

#Initialize the Variables.

decimal = int(input("Enter a Number Here:"))#print

the Entered Variable

print("The Conversion of Decimal Number",decimal,"is:")#decimal

number is converted into Binary print(bin(decimal),"in Binary")

#decimal number is converted into Octal

print(oct(decimal),"in Octal")

#decimal number is converted into Hexa Decimal

print(hex(decimal),"in Hexa Decimal")

OUTPUT:

Program-03 A

Aim: Demonstration of manipulation of strings using string method

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

#Create a Variable and Assign One Sentence

s = input("Enter a sentence: ")

#word,digits,uppercase,lowercase all are Firstly 0

w, d, u, l = 0, 0, 0, 0

#lenth word

l_w = s.split()

#create lenth word to assign w as variable

w = len(l_w)

#for character is String

for c in s:

#character is digits

if c.isdigit():

d = d + 1

#upper CASE

elif c.isupper():

u = u + 1

#lowerCase

```
elif c.islower():
```

```
l = l + 1
```

```
#Display the Number WOrd
```

```
print ("No of Words: ", w)
```

```
#Display the Number of Digits
```

```
print ("No of Digits: ", d)
```

```
#Display the Number of Upper Case
```

```
print ("No of Uppercase letters: ", u)
```

```
#Display the Number of Lower Case
```

```
print ("No of Lowercase letters: ", l)
```

OUTPUT:

Program-03 B

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

#Enter Strings Using Str Variables

str1 = input("Enter First String:\n")

str2 = input("Enter Second String\n")

#if string 2 is less than to string 1 assign values short

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

#Display the Similarity Two Strings

print("Similarity between two said String:")

print(matchCnt/ long)

OUTPUT:

Program-04 A

Aim: Discuss different collections like list, tuple and dictionary

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

INSERTION SORT PROGRAM

#Function Definition def

insertion_sort(alist):

#Start Range from 1 Upto Entered Elements are Ascending Order

for i in range(1, len(alist)):

temp = alist[i]

j = i - 1

while (j >= 0 and temp < alist[j]):

alist[j + 1] = alist[j]

j = j - 1

alist[j + 1] = temp

#Enter the List Of Items

alist = input('Enter The List of Numbers:').split()

```
alist = [int(x) for x in alist]
```

```
#function call
```

```
insertion_sort(alist)
```

OUTPUT:

MERGE SORT PROGRAM

```
def mergesort(list1):
```

```
    if len(list1) > 1:
```

```
        mid = len(list1) // 2
```

```
        left = list1[:mid]
```

```
        right = list1[mid:]
```

```
        mergesort(left)
```

```
        mergesort(right)
```

i = 0

j = 0

k = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

list1[k] = left[i]

i = i + 1

k = k + 1

else:

list1[k] = right[j]

j = j + 1

k = k + 1

while i < len(left): # if there is element left out in the left list

list1[k] = left[i]

i = i + 1

k = k + 1

while j < len(right): # if there is element left out in the right list

```
list1[k] = right[j]
```

```
j = j + 1
```

```
k = k + 1
```

```
list1 = input('enter the list of values to be sorted: ').split()
```

```
list1 = [int(x) for x in list1] # for every element in list1 we will call merge sort
```

```
mergesort(list1)
```

```
print(list1)
```

OUTPUT:

Program-04 B

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

```
class sol_Roman:
```

```
#Function Definition
```

```
def roman_to_integerNo(self, s):
```

```
    roman_no = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
```

```
    integer_no = 0
```

```
    for i in range(len(s)):
```

```
        if i > 0 and roman_no[s[i]] > roman_no[s[i - 1]]:
```

```
            integer_no += roman_no[s[i]] - 2 * roman_no[s[i - 1]]
```

```
    else:
```

```
        integer_no += roman_no[s[i]]
```

```
    return integer_no
```

```
#this is the A single One Line No Break Points here Mind Your Program Line
```

```
print("Roman Numerical to Integer is:",
```

```
sol_Roman().roman_to_integerNo(input("Enter the Roman Numericals:")))
```

OUTPUT:

Program-05 A

Aim: Demonstration of pattern recognition with and without using regular expression

a) Write a function called isphonenumbers () 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
```

```
#Function Definition For Is Number is True or False
```

```
def isphonenumbers(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
```

```
#function Definition For Check The Phone Number
```

```
def chkphonenumbers(numStr):
```

```
    ph_no_pattern = re.compile(r'^\d{3}-\d{3}-\d{4}$')
```



```
if ph_no_pattern.match(numStr):
```

```
    return True
```

```
else: else:
```

```
    return False
```

```
ph_num = input("Enter a phone number : ")
```

```
#without Using Regular Expressions
```

```
print("Without using Regular Expression")
```

```
if isphonenumber(ph_num):
```

```
    print("Valid phone number")
```

```
else:
```

```
    print("Invalid phone number")
```

```
#using Regular Expressions
```

```
print("Using Regular Expression")
```

```
if chkphonenumber(ph_num):
```

```
    print("Valid phone number")
```

```
else:
```

```
    print("Invalid phone number")
```

Program-05 B

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
```

```
# 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)
```

example.txt(Text File)

+917348878215

+919812569090

+916567976156

+917543679809

yuvi@gmail.com

Steps to Create This File

- This File Create In The Project File.
- Next Copy The Text On Above example.txt to which Created By inYour Project File.
- i.e You Are Created example.txt in Your Project
- Finally To Give The in Like This **with open('example.txt', 'r') as f:**
- Mind Your File Path is Should In Your Project File (if is Not Present InThe Project File It's Occurred Error Like File Not Found Error)

OUTPUT:

Program-06 A

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**

```
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")
```

Sample.txt(Text File)

this is phone number +917348878215

no phone number here

we have one +917348878215

we have an email aryahanumanthu@gmail.in and a number +917348878215

nothing of that sort here

**Better hope the life-inspector doesn't come around while you have your
life in such a mess.**

**You can create your own opportunities this week. Blackmail a senior
executive.**

Be different: conform.

Be cheerful while you are alive.

-- Phathotep, 24th Century B.C.

Q: How many journalists does it take to screw in a light bulb?

A: Three. One to report it as an inspired government program to bring light to the people, one to report it as a diabolical government plot to deprive the poor of darkness, and one to win a Pulitzer prize for reporting that Electric Company hired a light bulb-assassin to break the bulb in the first place.

Q: Why did the astrophysicist order three hamburgers?

A: Because he was hungry.

Q: Why haven't you graduated yet?

Steps to Create This File

- **This File Create In The Project File.**
- **Next Copy The Text On Above Sample.txt to which is Created By in Your Project File.**
- **i.e. You Are Created example.txt in Your Project**
- **Finally To Give The in Like This `with open('example.txt', 'r') as f:`**
- **Mind Your File Path is Should In Your Project File (if is Not Present In The Project File It's Occurred Error Like File Not Found Error)**

Program 06 B

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

```
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")

Steps To Create Directory

- **First in Project File Create Program File with Extension .py**
- **Next To Type or Copy and Paste the Above Program In**
Program06B
- **Next In the Project Folder Right click → Select New → Create**
Directory and Give Name As you Desired...

Program-07 A

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.

```
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:

Program 07 B

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):
```

```
        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:

Program-08

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 (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")
```

Program-09 A

Aim: Demonstration of working with excel spreadsheets and web scraping

a) 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.')

Program 09 B

b) 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:

Program-10A

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

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

```
from PyPDF2 import PdfWriter, PdfReader

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

pdf1 = open('birds.pdf', 'rb')
pdf2 = open('sample.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:

Program 10 B

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

```
import json

# Load the JSON data from file
with open('example.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}")
```

Example.json (This File Created On Your Project File Give name as

Example.json)

```
{
  "coord": {
    "lon": -73.99,
    "lat": 40.73
  },
  "weather": [
    {
      "id": 800,
```

```
"main": "Clear",
"description": "clear sky",
"icon": "01d"
},
"base": "stations",
"main": {
"temp": 10.45,
"feels_like": 12.74,
"temp_min": 14.44,
"temp_max": 16.11,
"pressure": 1017,
"humidity": 64
},
"visibility": 10000,
"wind": {
"speed": 8.63,
"deg": 180
},
"clouds": {
"all": 1
},
"dt": 1617979985,
"sys": {
"type": 1,
"id": 5141,
"country": "INDI",
"sunrise": 1617951158,
"sunset": 1618000213
},
"timezone": -14400,
"id": 5128581,
"name": "New York",
"cod": 200
}
```

OUTPUT:

MYCEM