**PYTHON PROGRAMMING LABORATORY (21CSL46)**

***INTERNAL EVALUATION SHEET***

|  |  |  |  |
| --- | --- | --- | --- |
| ***EVALUATION (MAX MARKS 50)*** | | | |
| **TEST**  **A** | **REGULAR EVALUATION**  **B** | **RECORD**  **C** | **TOTAL MARKS**  **A+B+C** |
| **20** | **20** | **10** | **50** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***R1: REGULAR LAB EVALUATION WRITE UP & PROGRAM EXECUTION RUBRIC (MAX MARKS 10)*** | | | | |
| **Sl. No.** | **Parameters** | **Good** | **Average** | **Needs**  **improvement** |
| **a.** | **Understanding of problem**  **(2 marks)** | Clear understanding of problem statement while designing and implementing the program (2) | Problem statement is understood clearly but few mistakes while designing and implementing program (1) | Problem statement is not clearly understood while designing the program (1) |
| **b.** | **Writing program**  **(2 marks)** | Program handles all possible conditions (2) | Average condition is defined and verified. (1) | Program does not handle possible conditions (1) |
| **c.** | **Design, implementation, and demonstration**  **(3 marks)** | Program follows syntax and semantics of C programming language. Demonstrates the complete knowledge of the program written (3) | Program has few logical errors, moderately demonstrates all possible concepts implemented in programs (2) | Syntax and semantics of C programming is not clear (1) |
| **d.** | **Result and documentation**  **(3 marks)** | Meticulous documentation and all conditions are taken care (3) | Acceptable documentation shown (2) | Documentation does not take care all conditions (1) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***R2: REGULAR LAB EVALUATION VIVA RUBRIC (MAX MARKS 10)*** | | | | | |
| **Sl. No.** | **Parameter** | **Excellent** | **Good** | **Average** | **Needs Improvement** |
| **a.** | **Conceptual understanding & Additional Programming**  **(10 marks)** | Answer 80% of the viva questions and execution of Additional programs listed (10) | Answers 60% of the viva questions and execution of Additional programs listed (7) | Answers 30% of the viva questions and execution of Additional programs listed(4) | Unable to relate the concepts (1) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***R3: RECORD EVALUATION RUBRIC (MAX MARKS 10)*** | | | | | |
| **Sl. No.** | **Parameter** | **Excellent** | **Good** | **Average** | **Needs Improvement** |
| **a.** | **Documentation**  **(10 marks)** | Meticulous record writing including program, comments and as per the guidelines mentioned (10) | Write up contains program, but comments are not included (8) | Write up contains only program (6) | Program written with few mistakes (5) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test 1 (8th Week)**  **(50 Marks)** | | | **Test 2 (14th Week)**  **(50 Marks)** | | |
| **Write up** | **Execution** | **Viva** | **Write up** | **Execution** | **Viva** |
| **10** | **20** | **20** | **10** | **20** | **10** |

# SYLLABUS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PYTHON PROGRAMMING LABORATORY (21CSL46)** | | | | |
| **Course Code** | | **21CSL46** | **CIE Marks** | 50 |
| **Number of Contact Hours/Week** | | 0:0:2:0 | **SEE Marks** | 50 |
| **Total Number of Lab Contact Hours** | | 24 | **Exam Hours** | 03 |
| **Credits –1** | | | | |
| **Descriptions(if any):** | | | | |
|  | | | | |
| **Programs List:** | | | | |
|  | | | | |
| 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.  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. | | | |
| 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.  b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions. | | | |
| 3 | 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 | | | | |
| 4 | 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. | | | |
| 5. | a) Write a function called isphonenumber () to recognize a pattern 415-555-4242without 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) | | | |
| 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  b) Write a python program to create a ZIP file of a particular folder which contains several files inside it. | | | |
| 7. | 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. | | | |
| 8 | 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. | | | |
| 9 | 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 | | | |
| 10 | 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 | | | |

**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 Objectives and Course Outcomes

|  |  |
| --- | --- |
| **COURSE LEARNING OBJECTIVES:** This course will enable students to | |
| 1. Demonstrate the use of IDLE or PyCharm IDE to create Python Applications | |
| 1. Using Python programming language to develop programs for solving real-world problems | |
| 1. Implement the Object-Oriented Programming concepts in Python. | |
| 1. Appraise the need for working with various documents like Excel, PDF, Word and Others | |
| 1. Demonstrate regular expression using python programming | |
| **Course Outcomes:** At the end of this course, students are able to: | |
| CO1 | Interpret the syntax and semantics of the Python programming language. |
| CO2 | Illustrate the data types and signify the usage of built-in functions of Python. |
| CO3 | Develop programs for string manipulation and file handling. |
| CO4 | Implement the Object Oriented Programming concepts in Python. |
| CO5 | Determine the need for scraping websites and solve the problems for a real time application |

**CO-PO-PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **COURSE**  **OUTCOMES** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
| **CO1** | 3 | 3 | 3 | 3 | 3 | 2 | 2 |  | 3 |  | 3 | 3 |  | 3 | 2 | 2 |
| **CO2** | 3 | 3 | 3 | 3 | 3 | 2 | 2 |  | 3 |  | 3 | 3 |  | 3 | 2 | 2 |
| **CO3** | 3 | 3 | 3 | 3 | 3 | 2 | 2 |  | 3 |  | 3 | 3 |  | 3 | 2 | 2 |
| **CO4** | 3 | 3 | 3 | 3 | 3 | 2 | 2 |  | 3 |  | 3 | 3 |  | 3 | 2 | 2 |
| **CO5** | 3 | 3 | 3 | 3 | 3 | 2 | 2 |  | 3 |  | 3 | 3 |  | 3 | 2 | 2 |

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

m1 = int(input("Enter marks for test1 : "))

m2 = int(input("Enter marks for test2 : "))

m3 = int(input("Enter marks for test3 : "))

if m1 <= m2 and m1 <= m3:

avgMarks = (m2+m3)/2

elif m2 <= m1 and m2 <= m3:

avgMarks = (m1+m3)/2

elif m3 <= m1 and m2 <= m2:

avgMarks = (m1+m2)/2

print("Average of best two test marks out of three test’s marks is", avgMarks);

**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.**

val = int(input("Enter a value : "))

str\_val = str(val)

if str\_val == str\_val[::-1]:

print("Palindrome")

else:

print("Not Palindrome")

for i in range(10):

if str\_val.count(str(i)) > 0:

print(str(i),"appears", str\_val.count(str(i)), "times");

**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 Fibonacci(n):

# Check if input is 0 then it will

# print incorrect input

if n < 0:

print("Incorrect input")

# Check if n is 0

# then it will return 0

elif n == 0:

return 0

# Check if n is 1,2

# it will return 1

elif n == 1 or n == 2:

return 1

else:

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

# Driver Program

n=int(input("enter the nth fibonacci series"))

print(Fibonacci(n))

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

def bin2Dec(val):

rev=val[::-1]

dec = 0

i = 0

for dig in rev:

dec += int(dig) \* 2\*\*i

i += 1

return dec

def oct2Hex(val):

rev=val[::-1]

dec = 0

i = 0

for dig in rev:

dec += int(dig) \* 8\*\*i

i += 1

list=[]

while dec != 0:

list.append(dec%16)

dec = dec // 16

nl=[]

for elem in list[::-1]:

if elem<= 9:

nl.append(str(elem))

else:

nl.append(chr(ord('A') + (elem -10)))

hex = "".join(nl)

return hex

num1 = input("Enter a binary number : ")

print(bin2Dec(num1))

num2 = input("Enter a octal number : ")

print(oct2Hex(num2))

**3. a) 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")

**3.b) 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)

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

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)

**4.b) Write a program to convert roman numbers in to integer values using dictionaries# Python program to convert Roman Numeral to Numbers**

# This function returns value of each Roman symbol

def value(r):

if (r == 'I'):

return 1

if (r == 'V'):

return 5

if (r == 'X'):

return 10

if (r == 'L'):

return 50

if (r == 'C'):

return 100

if (r == 'D'):

return 500

if (r == 'M'):

return 1000

return -1

def romanToDecimal(str):

res = 0

i = 0

while (i<len(str)):

# Getting value of symbol s[i]

s1 = value(str[i])

if (i + 1 <len(str)):

# Getting value of symbol s[i + 1]

s2 = value(str[i + 1])

# Comparing both values

if (s1 >= s2):

# Value of current symbol is greater

# or equal to the next symbol

res = res + s1

i = i + 1

else:

# Value of current symbol is greater

# or equal to the next symbol

res = res + s2 - s1

i = i + 2

else:

res = res + s1

i = i + 1

return res

# Driver code

print("Integer form of Roman Numeral is"),

print(romanToDecimal("MCMIV"))

**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**

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

**6.a) File Operations**

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

**Display the first N line of the fileFind 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(10):

print(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:

**6.b) 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")

**7 a) Inheritance : 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 = ""

11 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()

1. **b)Employee Details**

**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()

**8. a) Polymorphism and Inheritance**

**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")

**9. 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.')

Output :

**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()

**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('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)

**b) 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

}

**Sample Programs**

1. [**Python program to print "Hello Python"**](https://www.javatpoint.com/hello-python-program)

**print** ('Hello Python')

1. [**Python program to do arithmetical operations**](https://www.javatpoint.com/python-arithmetical-operations)

# Store input numbers:

num1 = input('Enter first number: ')

num2 = input('Enter second number: ')

# Add two numbers

sum = float(num1) + float(num2)

# Subtract two numbers

min = float(num1) - float(num2)

# Multiply two numbers

mul = float(num1) \* float(num2)

#Divide two numbers

div = float(num1) / float(num2)

# Display the sum

**print**('The sum of {0} and {1} is {2}'.format(num1, num2, sum))

# Display the subtraction

**print**('The subtraction of {0} and {1} is {2}'.format(num1, num2, min))

# Display the multiplication

**print**('The multiplication of {0} and {1} is {2}'.format(num1, num2, mul))

# Display the division

**print**('The division of {0} and {1} is {2}'.format(num1, num2, div))

3. [**Python program to find the area of a triangle**](https://www.javatpoint.com/python-area-of-triangle)

a = 5

b = 6

c = 7

# Uncomment below to take inputs from the user

# a = float(input('Enter first side: '))

# b = float(input('Enter second side: '))

# c = float(input('Enter third side: '))

# calculate the semi-perimeter

s = (a + b + c) / 2

# calculate the area

area = (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

print('The area of the triangle is %0.2f' %area)

**4.** [**Python program to swap two variables**](https://www.javatpoint.com/python-swap-two-variables)

#swapping of 2 numbers

p=int(input("enter the value of a"))

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

print('Before Swapping the value of p=',p,'and q=',q)

temp=p

p=q

q=temp

print('After swapping the value of p=',p,'and q=',q)

5. **Python program to find the sum and average of natural numbers up to n where n is provided by user.**

n=int(input("Enter upto which number you want sum and average"))

sum=0

for i in range(0,n+1):

sum=sum+i

avg=sum/n

print("Result of sum is",sum)

print("Result of Average",avg)

**6. WAP to find Factorial of a number using for loop**

fact=1

n=int(input("enter the value of n to find factorial of a given number"))

for i in range(1,n+1) :

fact=fact\*i

print(fact)

7. **WAP to find Factorial of a number using while loop**

fact=1

i=1

n=int(input("enter the value of n to find factorial of a given number"))

while i<=n :

fact=fact\*i

i=i+1

print(fact)

**8. WAP to find fibonacci series using Iterative:**

n = int(input("How many terms? "))

n1, n2 = 0, 1

count = 0

if n <= 0:

print("Please enter a positive integer")

elif n == 1:

print("Fibonacci sequence upto",n,":")

print(n1)

else:

print("Fibonacci sequence:")

while count < n:

print(n1)

next = n1 + n2

n1 = n2

n2 = next

count += 1

9. **WAP to find fibonacci series using recursion:**

def fib(n):

if n <= 1:

return n

else:

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

nterms = int(input("How many terms? "))

if nterms<= 0:

print("Plese enter a positive integer")

else:

print("Fibonacci sequence:")

for i in range(nterms):

print(fib(i))

10 **. WAP to find largest among three numbers, input by user**

n1=int(input("enter first number"))

n2=int(input("enter sec number"))

n3=int(input("enter third number"))

if n1>n2 and n1>n3 :

print("n1 is larger")

elif n2>n3 and n2>n1 :

print("n2 is larger")

else :

print("n3 is larger")

**11. WAP**  [**to print first ten programs using for loop .**](https://www.javatpoint.com/python-generate-random-number)

for i in range(1,11) :

print(i)

**12. WAP**  [**to print first ten programs using while loop.**](https://www.javatpoint.com/python-generate-random-number)

I=1

while i<=10:

print(i)

i=i+1

**13. WAP  [to check whether a person is eligible for voting.](https://www.javatpoint.com/python-generate-random-number)**

age=input()

type(age)

x=int(age)

if x>18 :

print("eligible")

else :

print("not ")

14. **WAP to print grades obtained by the students and print the appropriate message.**

marks=input()

type(marks)

x=int(marks)

if x>=90 and x<100:

print('distinction')

elif x>=80 and x<=90:

print("first")

else :

print("fail")

**15.WAP to find factorial of given number using for loop.**

fact=1

for i in range(1,6) :

fact\*=i

print(fact)

16.**WAP to find factorial of given number using while loop.**

fact=1

i=1

n=int(input())

while i<=n:

fact=fact\*i

i=i+1

print(fact)

17. **WAP to find factorial of given number using functions.**

def fact(n) :

fact=1

i=1

while i<=n :

fact\*=i

i=i+1

return fact

#print(fact)

n=int(input("enter the number to find factoral of a given number"))

print(fact(n))

18.**WAP to find gcd of 2 numbers.**

def gcd(a,b):

if b==0:

return a

else :

return gcd(b,a%b)

n1=int(input("enter the first number"))

n2=int(input("enter the second number"))

print(gcd(n1,n2))

19. [Python program to generate a random number](https://www.javatpoint.com/python-generate-random-number)

20. [Python program to convert kilometers to miles](https://www.javatpoint.com/python-convert-kilometers-to-miles).

21. [Python program to convert Celsius to Fahrenheit](https://www.javatpoint.com/python-convert-celsius-to-fahrenheit)

22. [Python program to display calendar](https://www.javatpoint.com/python-display-calendar)

23. [Python Program to Check if a Number is Positive, Negative or Zero](https://www.javatpoint.com/python-check-number-is-positive-negative-or-zero)

24. [Python Program to Check if a Number is Odd or Even](https://www.javatpoint.com/python-check-number-is-odd-or-even)

25. [Python Program to Check Leap Year](https://www.javatpoint.com/python-check-leap-year)

26. [Python Program to Check Prime Number](https://www.javatpoint.com/python-check-prime-number)

27. [Python Program to Print all Prime Numbers in an Interval](https://www.javatpoint.com/pyhton-print-all-prime-number-in-an-interval)

28. [Python Program to Find the Factorial of a Number](https://www.javatpoint.com/pyhton-factorial-number)

29. [Python Program to Display the multiplication Table](https://www.javatpoint.com/python-display-multiplication-table)

30. [Python Program to Print the Fibonacci sequence](https://www.javatpoint.com/python-print-the-fibonacci-sequence)

31.[Python Program to Check Armstrong Number](https://www.javatpoint.com/python-check-armstrong-number)

32.[Python Program to Find Armstrong Number in an Interval](https://www.javatpoint.com/python-armstrong-number)

**Python Additional Programs**

1. Python program to print "Hello Python"
2. Python program to do arithmetical operations
3. Python program to find the area of a triangle
4. Python program to solve quadratic equation
5. Python program to swap two variables
6. Python program to generate a random number
7. Python program to convert kilometers to miles
8. Python program to convert Celsius to Fahrenheit
9. Python program to display calendar
10. Python Program to Check if a Number is Positive, Negative or Zero
11. Python Program to Check if a Number is Odd or Even
12. Python Program to Check Leap Year
13. Python Program to Check Prime Number
14. Python Program to Print all Prime Numbers in an Interval
15. Python Program to Find the Factorial of a Number
16. Python Program to Display the multiplication Table
17. Python Program to Print the Fibonacci sequence
18. Python Program to Check Armstrong Number
19. Python Program to Find Armstrong Number in an Interval
20. Python Program to Find the Sum of Natural Numbers
21. Python Function Programs
22. Python Program to Find LCM
23. Python Program to Find HCF
24. Python Program to Convert Decimal to Binary, Octal and Hexadecimal
25. Python Program To Find ASCII value of a character
26. Python Program to Make a Simple Calculator
27. Python Program to Display Calendar
28. Python Program to Display Fibonacci Sequence Using Recursion
29. Python Program to Find Factorial of Number Using Recursion
30. Python Number Programs
31. Python program to check if the given number is a Disarium Number
32. Python program to print all disarium numbers between 1 to 100
33. Python program to check if the given number is Happy Number
34. Python program to print all happy numbers between 1 and 100
35. Python program to determine whether the given number is a Harshad Number
36. Python program to print all pronic numbers between 1 and 100
37. Python Array Programs
38. Python program to copy all elements of one array into another array
39. Python program to find the frequency of each element in the array
40. Python program to left rotate the elements of an array
41. Python program to print the duplicate elements of an array
42. Python program to print the elements of an array
43. Python program to print the elements of an array in reverse order
44. Python program to print the elements of an array present on even position
45. Python program to print the elements of an array present on odd position
46. Python program to print the largest element in an array
47. Python program to print the smallest element in an array
48. Python program to print the number of elements present in an array
49. Python program to print the sum of all elements in an array
50. Python program to right rotate the elements of an array
51. Python program to sort the elements of an array in ascending order
52. Python program to sort the elements of an array in descending order

**Viva Questions:**

1. What it the syntax of print function?

2. What is the usage of input function?

3. Define a variable.

4. What is type conversion?

5. Mention the data types in Python

6. What are the attributes of the complex datatype?

7. Mention a few escape sequences.

8. Define an expression

9. What is the usage of \*\* operator in Python?

10. Give the syntax of if else statement.

11. Give the syntax of for statement.

12. How is range function used in for?

13. Give the syntax of while statement.

14. What are multi way if statements?

15. How is random numbers generated?

16. Define a function.

17. Give the syntax of function.

18. What are the types of arguments in function.?

19. What is a recursive function?

20. What are anonymous functions?

21. What are default arguments?

22. What are variable length arguments?

23. What are keyword arguments?

24. Mention the use of map().

25. Mention the use of filter().

26. Mention the use of reduce().

27. Define a string.

28. How is string slicing done?

29. What is the usage of repetition operator?

30. How is string concatenation done using + operator>

31. Mention some string methods

32. How is length of a string found?

33. How is a string converted to its upper case?

34. `Differentiate isalpha() and isdigit().

35. What is the use of split()?

36. Define a file.

37. Give the syntax for opening a file.

38. Give the syntax for closing a file.

39. How is reading of file done?

40. How is writing of file done?

41. What is a list?

42. Lists are mutable-Justify.

43. How is a list created?

44. How can a list be sorted?

45. How are elements appended to the list?

46. How is insert() used in list?

47. What is the usage of pop() in list?

48. Define a tuple.

49. Are tuples mutable or immutable?

50. Mention the use of return statement.

51. What is a Boolean function?

52. How is main function defined?

53. What is a dictionary?

54. How are tuples created?

55. How is a dictionary created?

56. How to print the keys of a dictionary?

57. How to print the values of a dictionary?

58. How is del statement used?

59. Can tuple elements be deleted?

60. What is Python interpreter?

61. Why is Python called an interpreted language?

62. Mention some features of Python

63. What is Python IDLE?

64. Mention some rules for naming an identifier in Python.

65. Give points about Python Numbers.

66. What is bool datatype?

67. Give examples of mathematical functions.

68. What is string formatting operator?

69. Mention about membership operators in Python.

70. How is expression evaluated in Python?

71. What are the loop control statements in Python?

72. What is the use of break statement?

73. What is the use of continue statement?

74. What is the use of pass statement?

75. What is assert statement?

76. Differentiate fruitful function s and void functions.

77. What are required arguments ?

78. Differentiate pass by value and pass by reference.

79. Mention few advantages of function.

80. How is lambda function used?

81. What is a local variable?

82. What are global variables?

83. What are Python decorators?

84. Are strings mutable or immutable?

85. What is join()?

86. What is replace() method?

87. What is list comprehension?

88. Define multidimensional list.

89. How to create lists using range()?

90. What is swapcase() method?

91. What is linear search?

92. How is binary search done?

93. How is merge sort performed?

94. What is sorting?

95. How is insertion sort done?

96. How is selection sort done?

97. What are command line arguments?

98. Name some built in functions with dictionary.

99. What is an exception?

100. How is exception handled in python?