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

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

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

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

OUTPUT

```
Enter the marks in the first test: 45
Enter the marks in second test: 78
Enter the marks in third test: 23
The average of the best two test marks is:  61.5
```

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");
```

OUTPUT 1

```
 Enter a value : 1222221
Palindrome
1 appears 2 times
2 appears 5 times
```

OUTPUT 2

```
Enter a value : 12223
Not Palindrome
1 appears 1 times
2 appears 3 times
3    appears 1 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 fn(n):
            if n == 1:
                        return 0
            elif n == 2:
                        return 1
                else:
                        return fn(n-1) + fn(n-2)

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

        if num > 0:
                    print("fn(", num, ") = ",fn(num) , sep ="")
        else:
                    print("Error in input")
```

OUTPUT:

```
Enter a number : 0

Error in input


Enter a number : 7

fn(7) = 8
```

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))
```

OUTPUT:

```
Enter a binary number : 0001
1
Enter a octal number : 675
1BD
```

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

```
sentence = input("Enter a sentence : ")


wordList = sentence.split(" ")
print("This sentence has", len(wordList), "words")


digCnt = upCnt = loCnt = 0


for ch in sentence:
   if '0' <= ch <= '9':
        digCnt += 1
  elif 'A' <= ch <= 'Z':
        upCnt += 1
   elif 'a' <= ch <= 'z':
        loCnt += 1


print("This sentence has", digCnt, "digits", upCnt, "upper case letters", loCnt, "lower case letters")
```

OUTPUT:

```
Enter a sentence :  cse department at dsatm
This sentence has 4 words
This sentence has 0 digits 0 upper case letters 20 lower case letters
Enter a sentence : Cse department at DSATM 1234
This sentence has 5 words
This sentence has 4 digits 6 upper case letters 14 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)
```

OUTPUT:

```
Enter String 1
dsatmcse
Enter String 2
dsatmcse
Similarity between two said strings:
1.0
Enter String 1
cse department at dsatm
Enter String 2
ise department at dsatm
Similarity between two said strings:
0.9565217391304348
```

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

```
def insertion_sort(arr):
        for i in range(1,len(arr)):
                j=i
                while arr[j-1]>arr[j] and j>0:
                arr[j-1],arr[j]=arr[j],arr[j-1]
                j-=1
arr = []
def mergeSort(arr):
        if len(arr) > 1:
# Create sub_array2 ← A[start..mid] and sub_array2 ←A[mid+1..end]
        mid = len(arr)//2
        sub_array1 = arr[:mid]
        sub_array2 = arr[mid:]
# Sort the two halves
        mergeSort(sub_array1)
        mergeSort(sub_array2)
# Initial values for pointers that we use to keep track ofwhere we are in each array
        i = j = k = 0
# Until we reach the end of either start or end, pick largeramong# elements start and end and place them
in the correct positionin the sorted array
        while i < len(sub_array1) and j < len(sub_array2):
                if sub_array1[i] < sub_array2[j]:
                        arr[k] = sub_array1[i]
                        i += 1
                else:
                arr[k] = sub_array2[j]
```

```
    j += 1
    k += 1
# When all elements are traversed in either arr1 or arr2,# pick up the remaining elements and put in sorted array
    while i < len(sub_array1):
        arr[k] = sub_array1[i]
        i += 1
        k += 1
while j < len(sub_array2):
    arr[k] = sub_array2[j]
    j += 1
    k += 1
arr = []
# number of elements as input
n = int(input("Enter number of elements : "))
# iterating till the range
for i in range(0, n):
    ele = int(input())
    # adding the element
    arr.append(ele)
print("the numbers are")
print(arr)
print("insertion_sort")
insertion_sort(arr)
print(arr)
print("Merge_sort")
mergeSort(arr)
```

```
print(arr)
```

OUTPUT:

```
Unsorted List

[873, 781, 899, 241, 248, 452, 800, 716, 752, 583]

Sorting using Insertion Sort

[241, 248, 452, 583, 716, 752, 781, 800, 873, 899]


Unsorted List

[625, 71, 883, 877, 263, 872, 298, 781, 537, 97]

Sorting using Merge Sort

[71, 97, 263, 298, 537, 625, 781, 872, 877, 883]
```

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

```
s=input()
def romantointeger(s):
    d={'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000}
    ans=0
    for i in range(len(s)):
        if i+1 !=len(s) and d[s[i]]<d[s[i+1]]:
            ans=ans-d[s[i]]
        else:
            ans=ans+d[s[i]]
    return(ans)

print(romantointeger(s))
```

OUTPUT:

```
Enter a Roman Number : X

10
```

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:

```
Enter a phone number : 123-456-2345
Without using Regular Expression
Valid phone number
Using Regular Expression
Valid phone number

Enter a phone number : 123-fgt-7866
Without using Regular Expression
Invalid phone number
Using Regular Expression
Invalid phone number
```

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

```
        import re
        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)
```

```
INPUT                                               OUTPUT

EXAMPLE.TXT

cse   cse@gmail.com        +919876543210     cse@gmail.com     +919876543210

ise   ise@dsatm.edu.in     76545664          ise@dsatm.edu.in

ece   ece@dsatm.edu.in     12345678          ece@dsatm.edu.in

eee   eee@abc.efg.hij.com  12sd345g          eee@abc.efg
```

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.

```
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()
print(''.join(lineList))
word = input("Enter a word : ")
cnt = 0
for line in lineList:
    cnt += line.count(word)
print("The word", word, "appears", cnt, "times in the file")
```

OUTPUT

```
Enter the filename : example.txt
1 : this is phone number +918151894220
2 : no phone number here
3 : here we have one +829392938876
```

6b) 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")
```

OUTPUT

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

7. 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:

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

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

```
class Employee:
  def __init__(self):
          self.name = ""
          self.empId = ""
          self.dept = ""
           self.salary = 0
 def getEmpDetails(self):
          self.name = input("Enter Employee name : ")
          self.empId = input("Enter Employee ID : ")
           self.dept = input("Enter Employee Dept : ")
          self.salary = int(input("Enter Employee Salary : "))
def showEmpDetails(self):
           print("Employee Details")
          print("Name : ", self.name)
          print("ID : ", self.empId)
          print("Dept : ", self.dept)
          print("Salary : ", self.salary)
def updtSalary(self):
          self.salary = int(input("Enter new Salary : "))
           print("Updated Salary", self.salary)
e1 = Employee()
e1.getEmpDetails()
e1.showEmpDetails()
e1.updtSalary()
```

```
Enter Employee name : Sameer
```

OUTPUT

```
Enter Employee ID : A123
Enter Employee Dept : CSE
Enter Employee Salary : 85750

Employee Details
Name :  Sameer
ID :  A123
Dept :  CSE
Salary :  85750

Enter new Salary : 88800
Updated Salary 88800
```

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 PaliStr:

def __init__(self):
        self.isPali = False

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

class PaliInt(PaliStr):

    def __init__(self):
        self.isPali = False

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

        if val == rev:
            self.isPali = True
        else:
            self.isPali = False

        return self.isPali

st = input("Enter a string : ")

stObj = PaliStr()

if stObj.chkPalindrome(st):
    print("Given string is a Palindrome")
else:
    print("Given string is not a Palindrome")

val = int(input("Enter a integer : "))
```

```
intObj = PaliInt()
if intObj.chkPalindrome(val):
   print("Given integer is a Palindrome")
else:
    print("Given integer is not a Palindrome")
```

OUTPUT:

```
Enter a string : csedsatm
Given string is not a Palindrome
Enter a integer : 123321
Given integer is 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:

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

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

```
from openpyxl import Workbook
from openpyxl.styles import Font
wb = Workbook()
sheet = wb.active
sheet.title = "Language"
wb.create_sheet(title = "Capital")
lang = ["Kannada", "Telugu", "Tamil"]
state = ["Karnataka", "Telangana", "Tamil Nadu"]
capital = ["Bengaluru", "Hyderabad", "Chennai"]
code =['KA', 'TS', 'TN']
sheet.cell(row = 1, column = 1).value = "State"
sheet.cell(row = 1, column = 2).value = "Language"
sheet.cell(row = 1, column = 3).value = "Code"
ft = Font(bold=True)
for row in sheet["A1:C1"]:
  for cell in row:
              cell.font = ft
for i in range(2,5):
              sheet.cell(row = i, column = 1).value = state[i-2]
            sheet.cell(row = i, column = 2).value = lang[i-2]
             sheet.cell(row = i, column = 3).value = code[i-2]
wb.save("demo.xlsx")
sheet = wb["Capital"]
sheet.cell(row = 1, column = 1).value = "State"
sheet.cell(row = 1, column = 2).value = "Capital"

sheet.cell(row = 1, column = 3).value = "Code"

ft = Font(bold=True)

for row in sheet["A1:C1"]:

            for cell in row:

                        cell.font = ft

for i in range(2,5):

           sheet.cell(row = i, column = 1).value = state[i-2]

           sheet.cell(row = i, column = 2).value = capital[i-2]

            sheet.cell(row = i, column = 3).value = code[i-2]

wb.save("demo.xlsx")

srchCode = input("Enter state code for finding capital ")

for i in range(2,5):

             data = sheet.cell(row = i, column = 3).value

  if data == srchCode:

print("Corresponding capital for code", srchCode, "is", sheet.cell(row = i, column = 2).value)
sheet = wb["Language"]

srchCode = input("Enter state code for finding language ")

for i in range(2,5):

               data = sheet.cell(row = i, column = 3).value

   if data == srchCode:

 print("Corresponding language for code", srchCode, "is", sheet.cell(row = i, column = 2).value)

wb.close()
```

OUTPUT

```
Enter state code for finding capital KA

Corresponding capital for code KA is Bengaluru
```

10 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)
```

OUTPUT

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

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

Part -A

```
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}")
```

part B

# In note pad file type the below code and save as .json in the same folder

```
  "coord": {
    "lon": -73.99,
    "lat": 40.73
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01d"
    }
```

```
],
"base": "stations",
"main": {
  "temp": 15.45,
  "feels_like": 12.74,
  "temp_min": 14.44,
  "temp_max": 16.11,
  "pressure": 1017,
  "humidity": 64
},
"visibility": 10000,
"wind": {
  "speed": 4.63,
  "deg": 180
},
"clouds": {
  "all": 1
},
"dt": 1617979985,
"sys": {
  "type": 1,
  "id": 5141,
  "country": "US",
  "sunrise": 1617951158,
  "sunset": 1618000213
},
"timezone": -14400,
  "id": 5128581,
  "name": "New York",
  "cod": 200
}
```

OUTPUT

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

PART- B QUESTIONS

1. Write a Python program to find the biggest of three numbers
   i)without taking input from the user
   ii)taking input from the user
2. Write a python program using functions to add two numbers
3. Write a python program
   i)to compare two strings
   ii)to join two strings
4. Write a python program using list to show the following slicing operation
   a)i)items from index 2 to index 4
   ii)items from index 5 to end
   iii)items beginning to end
   b)Write a program to show empty tuple,tuple with integers,tuple with different data typesanf nested tuple
5. Write a program to read the first line using readline()
6. Write a program in python to copy all the contents of one file to another file in upper case
7. Write a program in Python to demonstrate inheritance property for the Student class
8. Write a program in python to demonstrate polymorphism
9. Write a program to show the exception handling in Python
10. Show the working of Nested for loop in lists

1.Write a Python program to find the biggest of three numbers

i)without taking input from the user

ii)taking input from the user

```
# Python program to find the largest number among the three input numbers


# change the values of num1, num2 and num3

# for a different result

num1 = 10

num2 = 14

num3 = 12


# uncomment following lines to take three numbers from user

#num1 = float(input("Enter first number: "))

#num2 = float(input("Enter second number: "))

#num3 = float(input("Enter third number: "))


if (num1 >= num2) and (num1 >= num3):

   largest = num1

elif (num2 >= num1) and (num2 >= num3):

   largest = num2

else:

   largest = num3


print("The largest number is", largest)
```

Result: The largest number is 14

2.Write a python program using functions to add two numbers

```
# function with two arguments
def add_numbers(num1, num2):
    sum = num1 + num2
    print("Sum: ",sum)

# function call with two values
add_numbers(5, 4)
```

Output: Sum: 9

3.Write a python program   i)to compare two strings ii)to join two strings

```
i) str1 = "Hello, world!"
str2 = "I love Python."
str3 = "Hello, world!"
# compare str1 and str2
print(str1 == str2)
# compare str1 and str3
print(str1 == str3)
```

output False

```
        True
```

```
ii) greet = "Hello, "
  name = "Jack"
# using + operator
result = greet + name
print(result)
```

Output: Hello, Jack

4.Write a python program using list to show the following slicing operation

a)i)items from index 2 to index 4

ii)items from index 5 to end

iii)items beginning to end

```
# List slicing in Python
my_list = ['p','r','o','g','r','a','m','i','z']
# items from index 2 to index 4
print(my_list[2:5])
# items from index 5 to end
print(my_list[5:])
# items beginning to end
print(my_list[:])
```

b)Write a program to show empty tuple,tuple with integers,tuple with different data typesanf nested tuple

```
 # Different types of tuples
# Empty tuple
my_tuple = ()
print(my_tuple)
# Tuple having integers
my_tuple = (1, 2, 3)
print(my_tuple)
# tuple with mixed datatypes
my_tuple = (1, "Hello", 3.4)
print(my_tuple)
# nested tuple
my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
print(my_tuple)
```

c)Write a python program to implement dictionary to print States and their capitals

5.Write a program to read the first line using readline()

```
myfile = open("demo.txt", "r")
myline = myfile.readline()
print(myline)
myfile.close()
```

demo.txt

```
Testing - FirstLine
Testing - SecondLine
Testing - Third Line
Testing - Fourth Line
Testing - Fifth Line
```

Output:

Testing – FirstLine

6 .Write a program in python to copy all the contents of one file to another file in upper case

```
To open the first file in read mode
f1 = open("sample file 1.txt", "r")

# To open the second file in append mode
f2 = open("sample file 2.txt", "a")

# For loop to traverse through the file
for line in f1:

    # Writing the content of the first
    # file to the second file

    # Using upper() function
    # to capitalize the letters
    f2.write(line.upper())
```

7 a)Write a proram in Python to demonstrate inheritance property for the Student class

```
class Person:
  def __init__(self, fname, lname):
    self.firstname = fname
    self.lastname = lname

  def printname(self):
    print(self.firstname, self.lastname)

class Student(Person):
  def __init__(self, fname, lname, year):
    super().__init__(fname, lname)
    self.graduationyear = year

x = Student("Mike", "Olsen", 2019)
print(x.graduationyear)
```

8.Write a program in python to demonstrate polymorphism

```
# A simple Python function to demonstrate
# Polymorphism

def add(x, y, z = 0):
            return x + y+z
```

```
# Driver code
print(add(2, 3))
print(add(2, 3, 4))
```

9.Write aprogram to show the exception handling in Python

```
# Program to handle multiple errors with one
# except statement
# Python 3

def fun(a):
    if a < 4:

        # throws ZeroDivisionError for a = 3
        b = a/(a-3)

    # throws NameError if a >= 4
    print("Value of b = ", b)

try:
    fun(3)
    fun(5)

# note that braces () are necessary here for
# multiple exceptions
except ZeroDivisionError:
    print("ZeroDivisionError Occurred and Handled")
except NameError:
    print("NameError Occurred and Handled")
```

output:

```
ZeroDivisionError Occurred and Handled
```

10.Show the working of Nested for loop in lists

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
  for y in fruits:
    print(x, y)
```

output:

```
red apple
red banana
red cherry
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry
```

1. Why python is called Object oriented language
2. What are the characteristics of Python?
3. Who is the founder of Python?
4. Why List is called Sequential and mutable data type?
5. Give examples of immutable data types
6. Give the differences between Python and C languages
7. Give the differences between Dictonary and Tuple.
8. Explain Escape characters in Python.
9. How to give multiline comment in python?
10. What are the supported data types in Python?
11. What is the output of print str[0] if str = 'Hello World!'?
12. What is the output of print str[2:5] if str = 'Hello World!'?
13. What is the output of print str * 2 if str = 'Hello World!'?
14. How will you convert a string to an int in python?

15. What is the purpose of // operator?
16. Explain polymormishm and Encapsulation in Python
17. Give the difference between shallow () and deep()
18. Discuss Inheritance
19. Explain class concepts in Python
20. Find the output for the following code

```
a)while count < nterms:
      print(n1)
      nth = n1 + n2
      # update values
      n1 = n2
      n2 = nth
      count += 1

b)languages = ['Swift', 'Python', 'Go', 'JavaScript']

# run a loop for each item of the list
for language in languages:
    print(language)
```