**PROGRAMS**

**Class 1:**

**1.WAP to print Hello World**

print(“Hello World”)

**2.WAP implements Airthematic operators.**

a=int(input(“Enter first number”))

b=int(input(“Enter second number”))

print(“Addition”,a+b)

print(“Subtract”, a-b)

print(“Divide float”, a/b)

print(“Divide integer value”, a//b)

print(“Modulo”, a/b)

print(“Exponent”, a\*\*b)

**3.WAP implements Bitwise operators.**

a=int(input(“Enter first number”))

b=int(input(“Enter second number”))

print(“Addition”,a&b)

print(“Subtract”, a|b)

print(“Divide float”, a<<b)

print(“Divide integer value”, a>>b)

print(“Modulo”, a^b)

print(“Exponent”, ~b)

**4.WAP to take a character from user and convert it into its equivalent ASCII**

a=65

print(chr(a))

print(ord('a'))

**5.WAP to find out AVG of 5 numbers**

M1=int(input(“Enter first number”))

M2=int(input(“Enter second number”))

M3=int(input(“Enter third number”))

M4=int(input(“Enter fourth number”))

M5=int(input(“Enter fifth number”))

AVG=(M1+M2+M3+M4+M5)/5

print(“Avg marks:”AVG)

**6.WAP to calculate SI**

p=int(input(“Enter principal amount”))

r=int(input(“Enter rate”))

t=int(input(“Enter time”))

si=(p\*r\*t)/100

print(“Simple Interest is:”,si

**7.WAP to calculate CI**

p=int(input(“Enter principal amount”))

r=int(input(“Enter rate ”))

t=int(input(“Enter time”))

n=int(input(“Number of time interest compounded per year”))

A=p(1+r/n)\*\*t

Ci=A-p

print(“Compound interest::“,ci)

**Class2 :**

**1.WAP to convert temperature in different scale**

far=int(input(“Enter temperature in Fahrenheit”))

c=(F-32)\*5/9

print(“Temperature in Fahrenheit::”,far)

print(“Temperature in celcius::”,c)

**2.WAP to take a number from user and check whether it is divisible by 9 or not**

n=int(input(“Enter the number”))

if(n%9==0):

print(“Number is divisible by 9”)

else:

print(“Number is not divisible by 9”)

**3.WAP to take A number from user and check whether the number is even or odd**

n=int(input(“Enter the number”))

if(n%2==0):

print(“Number is EVEN”)

else:

print(“Number is ODD”)

**4.WAP to check whether the number is even or odd without using % operator**

n=int(input(“Enter the number”))

if(nAND1==0):

print(“Number is divisible by 9”)

else:

print(“Number is not divisible by 9”)

**5. Create a python script that takes amount from user and calculate paid amount after the discount. Discounts are given by the following:**

**a)Amount is less than 2000 - No discount**

**b) Amount is greater than or equal to 2000 - 2%**

**c) Amount is greater than or equal to 3000 - 3%**

**d) Amount is greater than or equal to 4000 - 4%**

**e) Amount is greater than or equal to 5000 - 10%**

**f) Amount is greater than or equal to 10000 - 30%**

amt=int(input(“Enter your amount”))

if(amt<2000):

print(“No discount”)

elif(amt<3000):

printf(amt-amt\*0.02)

elif(amt<4000):

print(amt-amt\*0.03)

elif (amt<5000):

printf(amt-amt\*0.04)

elif(amt<5000):

print(amt-amt\*0.04)

elif(5000<=amt<10000):

print(amt-amt\*0.1)

else:

print(amt-amt\*.3)

**6. WAP that take user name and password from user and verify login criteria**

Uname=input(“Enter username”)

Pwd=input(“Enter passsword”)

If(Uname==yourname AND pwd==your password):

Print(“Login sucessfull”)

else:

print(“login unsuccesfull”)

**7**.**WAP that display difference between two numbers**

n1=int(input(“Enter first number”))

n2= int(input(“Enter second number”))

if(n1>n2):

print(n1-n2)

else:

print(n2-n1)

**Class3:**

**1.WAP that takes two number from user and find greater between them**

n1=int(input(“Enter first number”))

n2= int(input(“Enter second number”))

if(n1>n2):

print(“n1 is greator”)

else:

print(“n2 is greator”)

**2.WAP to perform swapping of two numbers**

a=int(input(“Enter a number”))

b=int(input(“Enter second number”))

print(“Elements before swapping”,a,b)

t=a

a=b

b=t

print(“Elements after swapping”,a,b)

**3.WAP to find out number is prime or not.**

n=int(input("Enter any number"))

c=0

i=1

while(i<=n/2):

if(n%i==0):

c=c+1

i=i+1

if(c==1):

print(n," is prime")

else:

print(n,"is not prime")

**4.WAP to take a range from user and check out which one is prime and which is not**

a=int(input("Enter any number"))

b=int(input("Enter any number"))

while a<b:

c=0

i=1

while(i<=a/2):

if(a%i==0):

c=c+1

i=i+1

if(c==1):

print(a," is prime")

else:

print(a,"is not prime")

a=a+1

**5.WAP to find out fibonacii series of user dependent range**

range=int(input("Enter the end range of series"))

f0=0

f1=1

print("fibonacii series:",f0,f1,end=" ")

i=1

while i<=range:

s=f1+f0

f0=f1

f1=s

i=i+1

print(s,end=" ")

**6.WAP to draw following pattern:**

**1**

**12**

**123**

**1234**

i=1

s=0

while i<5:

s=s\*10+i

print(s)

i=i+1

**7. \*\*\*\***

**\*\*\***

**\*\***

**\***

i=1

while i<=4:

j=1

while j<=4:

if(j<i):

print(" ",end="")

else:

print("\*",end="")

j=j+1

print("")

i=i+1

**Class 4th:**

**1. \*\*\*\***

**123**

**\*\***

**1**

i=1

while i<=4:

j=1

while j<=4:

if(j<i):

print(" ",end="")

elif i%2==0:

if i==4 and j==4:

print(1)

else:

print(k,end="")

else:

print("\*",end="")

k=j

j=j+1

print("")

i=i+1

**2. \***

**\*\***

**\*\*\***

**\*\*\*\***

i=1

while i<5:

j=1

while j<5:

if(j<5-i):

print(" ",end="")

else:

print("\*",end="")

j=j+1

print("")

i=i+1

**3. Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700 (both included)**

nl=[]

for x inrange(1500,2701):

if(x%7==0)and(x%5==0):

nl.append(str(x))

print(','.join(nl))

**4.Write a Python program to construct the following pattern, using a nested for loop.**

**\* \*   
\* \* \*   
\* \* \* \*   
\* \* \* \* \*   
\* \* \* \*   
\* \* \*   
\* \*   
\***

n=5;

foriinrange(n):

for j inrange(i):

print('\* ', end="")

print('')

foriinrange(n,0,-1):

for j inrange(i):

print('\* ', end="")

print('')

**5. Write a Python program to construct the following pattern, using a nested for loop.**

**\***

**\*\***

**\*\*\***

**\*\*\*\***

i=1

while i<5:

j=1

while j<5:

if(j<5-i):

print(" ",end="")

else:

print("\*",end="")

j=j+1

print("")

i=i+1

**6. Write a Python program to construct the following pattern, using a nested for loop.**

**\***

**\* \* \***

**\* \* \* \* \***

**\* \* \* \* \* \* \***

i=1

while i<=4:

j=i

while j<=3:

print(" ",end="")

j=j+1

k=1

while k<=2\*i-1:

print("\*",end=" ")

k=k+1

print(" ")

i=i+1

**7.Write a Python program to construct the following pattern, using a nested for loop.**

\* \* \* \* \*

\* \* \*

\*

i=1

while i<=4:

j=1

while j<=7:

if(j<=7-i):

if(j<=i):

print(" ",end=' ')

else:

print("\*",end=' ')

j=j+1

print(" ")

i=i+1

**class5th:**

**1.WAP that take inputs as in lower case and print in upper case**

i=0

str=input("Enter any string")

n=len(str)

while i<n:

a=str[i]

print(chr(ord(a)+32),end="")

i=i+1

**2.WAP to make toggle case**

i=0

str=input("Enter any string")

n=len(str)

while i<n:

a=str[i]

if ord(a)>=97 and ord(a)<=122:

print(chr(ord(a)-32),end="")

eliford(a)>=65 and ord(a)<=90:

print(chr(ord(a)+32),end="")

i=i+1

**3.WAP to make following patern of string**

**C**

**CO**

**COM**

**COMP**

**COMPU**

**COMPUT**

**COMPUTE**

**COMPUTEC**

str=input("Enter any string")

n=len(str)

i=0

while i!=n:

j=0

while j<=i:

print(str[j],end="")

j=j+1

print(" ")

i=i+1

**4.WAP to make sentence case**

i=0

str=input("Enter any string")

n=len(str)

a=str[0]

print(chr(ord(a)-32)+str[1:n+1])

**5.Write a Python program to count the number of characters (character frequency) in a string.**

print("Enter 'x' for exit.")

string = input("Enter any string to count character: ")

if string == 'x':

exit()

else:

char = input("Enter a character to count to count from above string: ")

val = string.count(char)

print("Total = ",val)

**6.Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.**

def string\_both\_ends(str):

if len(str) < 2:

return ''

return str[0:2] + str[-2:]

print(string\_both\_ends('Computec'))

print(string\_both\_ends('Co'))

print(string\_both\_ends('C'))

**7.Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself.**

def change\_char(str1):

char = str1[0]

length = len(str1)

str1 = str1.replace(char, '$')

str1 = char + str1[1:]

return str1

print(change\_char('restart'))

**Class 7th**

**1.Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.**

def chars\_mix\_up(a, b):

new\_a = b[:2] + a[2:]

new\_b = a[:2] + b[2:]

return new\_a + ' ' + new\_b

print(chars\_mix\_up('abc', 'xyz'))

**2.WAP to perfoerm all the operations in list**

l1=[10,20,30,40,50,50,20,50]

l1.append(4)

print(l1)

l2=[6,5,7]

l1.extend(l2)

print(l1)

l1.sort()

print(l1)

a=l1.pop()

print(a)

print(l1)

l1.reverse()

print(l1)

l1.insert(2,3)

print(l1)

**3.WAP to make a list of 20 numbers and add the elements n them and multiply them**

l=[]

i=0

s=0

m=1

while i<20:

l.append(int(input("Enter number")))

s=s+l[i]

m=m\*l[i]

i=i+1

print("sum",s)

print("multiply",m)

**4.WAP to make sum of two lists**

i=0

l1=[]

l2=[]

l3=[]

n=int(input("Enter how many number you want to enter"))

while i<n:

l1.append(int(input("Enter element of firstlist")))

l2.append(int(input("Enter element of second list")))

l3.append(l1[i]+l2[i])

i=i+1

print(l3)

**5.WAP to find out Even numbers in list**

i=0

l1=[]

n=int(input("Enter how many number you want to enter"))

while i<n:

l1.append(int(input("Enter element of list")))

i=i+1

i=0

while i<n:

if l1[i]%2==0:

print("Even number:",l1[i])

else:

print("Odd number:",l1[i])

i=i+1

**6.WAP to perform searching of elements in list and if not found then add them.**

i=0

l1=[10,20,30,40,50]

s=0

element=int(input("Enter any number you want to search"))

for j in l1:

if j==element:

s=s+1

new=int(input("Enter new number you want to replace"))

l1[i]=new

i=i+1

if s>=1:

print("Element found and replaced by new element list is:",l1)

else:

print("element not found list is:",l1)

**7.Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.**

defmatch\_words(words):

ctr =0

for word in words:

iflen(word)>1and word[0]== word[-1]:

ctr +=1

return ctr

print(match\_words(['abc','xyz','aba','1221']))

**Class 8th:**

**1.Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples.**

def last(n):

return n[-1]

def sort\_list\_last(tuples):

return sorted(tuples, key=last)

print(sort\_list\_last([(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]))

**2.Write a Python program to create a tuple.**

x = ()

print(x)

tuplex = tuple()

print(tuplex)

**3.Write a Python program to create a tuple with different data types**

tuplex=("tuple",False,3.2,1)

print(tuplex)

**4.Write a Python program to create a tuple with numbers and print one item.**

tuplex=5,10,15,20,25

print(tuplex)

tuplex=5,

print(tuplex)

**5.Write a Python program to add an item in a tuple**

tuplex=(4,6,2,8,3,1)

print(tuplex)

tuplex=tuplex+(9,)

print(tuplex)

tuplex=tuplex[:5]+(15,20,25)+tuplex[:5]

print(tuplex)

listx=list(tuplex)

listx.append(30)

tuplex=tuple(listx)

print(tuplex)

**6.** **Write a Python program to convert a tuple to a string.**

tup=('e','x','e','r','c','i','s','e','s')

str=''.join(tup)

print(str)

**7.Write a Python program to create the colon of a tuple**

from copy importdeepcopy

tuplex=("HELLO",5,[],True)

print(tuplex)

tuplex\_clone=deepcopy(tuplex)

tuplex\_clone[2].append(50)

print(tuplex\_clone)

print(tuplex)

**class 9th:**

**1.Write a Python program to find the repeated items of a tuple.**

tuplex= (2,4,5,6,2,3,4,4,7)

print(tuplex)

count =tuplex.count(4)

print(count)

**2.Write a Python script to sort (ascending and descending) a dictionary by value**

import operator

d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}

print('Original dictionary : ',d)

sorted\_d = sorted(d.items(), key=operator.itemgetter(0))

print('Dictionary in ascending order by value : ',sorted\_d)

sorted\_d = sorted(d.items(), key=operator.itemgetter(0),reverse=True)

print('Dictionary in descending order by value : ',sorted\_d)

**3.Write a Python script to add a key to a dictionary.**

d ={0:10,1:20}

print(d)

d.update({2:30})

print(d)

**4.Write a Python script to concatenate following dictionaries to create a new one.**

dic1={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

dic4 ={}

for din(dic1, dic2, dic3):

dic4.update(d)

print(dic4)

**5.Write a Python script to check if a given key already exists in a dictionary**

d ={1:10,2:20,3:30,4:40,5:50,6:60}

defis\_key\_present(x):

if x in d:

print('Key is present in the dictionary')

else:

print('Key is not present in the dictionary')

is\_key\_present(5)

is\_key\_present(9)

**6.Write a Python program to iterate over dictionaries using for loops.**

d = {'x': 10, 'y': 20, 'z': 30}

for dict\_key, dict\_value in d.items():

print(dict\_key,'->',dict\_value)

**7.Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x).**

n=int(input("Input a number "))

d =dict()

for x inrange(1,n+1):

d[x]=x\*x

print(d)

**class 10th:**

**1.Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys**

d=dict()

for x inrange(1,16):

d[x]=x\*\*2

print(d)

**2.Write a Python script to merge two Python dictionaries**

d1 ={'a':100,'b':200}

d2 ={'x':300,'y':200}

d = d1.copy()

d.update(d2)

print(d)

**3.Write a Python function to find the Max of three numbers.**

defmax\_of\_two( x, y ):

if x > y:

return x

return y

defmax\_of\_three( x, y, z ):

returnmax\_of\_two( x,max\_of\_two( y, z ))

print(max\_of\_three(3, 6, -5))

**4.Write a Python function to sum all the numbers in a list**

defsum(numbers):

total =0

for x in numbers:

total += x

return total

print(sum((8, 2, 3, 0, 7)))

**5.Write a Python function to sum all the numbers in a list.**

defsum(numbers):

total =0

for x in numbers:

total += x

return total

print(sum((8,2,3,0,7)))

**6.Write a Python program to reverse a string.**

defstring\_reverse(str1):

rstr1 =''

index =len(str1)

while index >0:

rstr1 += str1[ index -1]

index = index -1

return rstr1

print(string\_reverse('1234abcd'))

**7.Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.**

deffactorial(n):

if n ==0:

return1

else:

return n \* factorial(n-1)

n=int(input("Input a number to compute the factiorial : "))

print(factorial(n))

**class 11.**

**1.Write a Python function to check whether a number is perfect or not.   
According to Wikipedia : In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).  
*Example* : The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: ( 1 + 2 + 3 + 6 ) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128**

defperfect\_number(n):

sum=0

for x inrange(1, n):

if n % x ==0:

sum+= x

returnsum== n

print(perfect\_number(6))

**2.Write a Python function that checks whether a passed string is palindrome or not.   
Note: A palindrome is a word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.**

def isPalindrome(string):

left\_pos = 0

right\_pos = len(string) - 1

while right\_pos>= left\_pos:

if not string[left\_pos] == string[right\_pos]:

return False

left\_pos += 1

right\_pos -= 1

return True

print(isPalindrome('aza'))

**3.Write a Python function to check whether a string is a pangram or not.  
Note : Pangrams are words or sentences containing every letter of the alphabet at least once.  
For example : "The quick brown fox jumps over the lazy dog"**

import string, sys

defispangram(str1, alphabet=string.ascii\_lowercase):

alphaset=set(alphabet)

returnalphaset<=set(str1.lower())

print(ispangram('The quick brown fox jumps over the lazy dog'))

**4.WAP to find current time of your system.**

importtime

print(time.asctime())

**5.WAP to find out calendar of any month any year.**

import calendar

calendar.month(2018,9)

calendar.calendar(2018)

**6.WAP to find out factorial of any number using math module.**

import math

n=int(input(“Enter any number”))

z=math.factorial(n)

print(z)

**7.WAP to make user defined module.**

**defsquare**(x):

**return**x**\***x

**defcube**(x):

**return**x**\***x**\***x

**defsquare**(x):

**return**x**\***x

**defcube**(x): save it by num.py

**return**x**\***x**\***x

next editor:

import num

num.square(10)

num.cube(5)

**Class 12:**

**1.Write a Python program to read an entire text file.**

deffile\_read(fname):

txt =open(fname)

print(txt.read())

file\_read('test.txt')

**2.Write a Python program to read first n lines of a file.**

def file\_read\_from\_head(fname, nlines):

from itertools import islice

with open(fname) as f:

for line in islice(f, nlines):

print(line)

file\_read\_from\_head('test.txt',2)

**3.Write a Python program to read a file line by line and store it into a list.**

deffile\_read(fname):

withopen(fname)as f:

content\_list=f.readlines()

print(content\_list)

file\_read(\'test.txt\')

**4.Write a Python program to read a file line by line store it into a variable**

deffile\_read(fname):

withopen(fname,"r")asmyfile:

data=myfile.readlines()

print(data)

file\_read('test.txt')

**5.Write a python program to find the longest words.**

deflongest\_word(filename):

withopen(filename,'r')asinfile:

words =infile.read().split()

max\_len=len(max(words, key=len))

return[word for word in words iflen(word)==max\_len]

print(longest\_word('test.txt'))

**6.Write a Python program to count the frequency of words in a file.**

from collections import Counter

defword\_count(fname):

withopen(fname)as f:

return Counter(f.read().split())

print("Number of words in the file :",word\_count("test.txt"))

**7.Write a Python program to write a list to a file**

color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

with open('abc.txt', "w") as myfile:

for c in color:

myfile.write("%s\n" % c)

content = open('abc.txt')

print(content.read())

**Class 13:**

**1.Write a Python program to convert degree to radian.**

pi=22/7

degree =float(input("Input degrees: "))

radian = degree\*(pi/180)

print(radian)

**2.Write a Python program to calculate surface volume and area of a sphere.**

pi=22/7

radian = float(input('Radius of sphere: '))

sur\_area = 4 \* pi \* radian \*\*2

volume = (4/3) \* (pi \* radian \*\* 3)

print("Surface Area is: ", sur\_area)

print("Volume is: ", volume)

**3.Write a Python program to calculate arc length of an angle**

def arclength():

pi=22/7

diameter = float(input('Diameter of circle: '))

angle = float(input('angle measure: '))

if angle >= 360:

print("Angle is not possible")

return

arc\_length = (pi\*diameter) \* (angle/360)

print("Arc Length is: ", arc\_length)

arclength()

**4.Write a Python program to find the smallest multiple of the first n numbers. Also, display the factors**

defsmallest\_multiple(n):

i= n \*2

factors =[number for number inrange(n,1,-1)if number \*2> n]

print(factors)

whileTrue:

for a in factors:

ifi% a !=0:

i+= n

break

if(a == factors[-1]andi% a ==0):

returni

print(smallest\_multiple(13))

print(smallest\_multiple(11))

**5.Write a Python program to returns sum of all divisors of a numbers**

defsum\_div(number):

divisors =[1]

foriinrange(2, number):

if(number %i)==0:

divisors.append(i)

return divisors

print(sum\_div(8))

print(sum\_div(12))

**6.WAP to generate fibonacii series.**

def recur\_fibo(n):

if n <= 1:

return n

else:

return(recur\_fibo(n-1) + recur\_fibo(n-2))

nterms = 10

# uncomment to take input from the user

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

# check if the number of terms is valid

if nterms<= 0:

print("Plese enter a positive integer")

else:

print("Fibonacci sequence:")

for i in range(nterms):

print(recur\_fibo(i))

**7.WAP to add two matrices**

X = [[12,7,3],

[4 ,5,6],

[7 ,8,9]]

Y = [[5,8,1],

[6,7,3],

[4,5,9]]

result = [[0,0,0],

[0,0,0],

[0,0,0]]

# iterate through rows

for i in range(len(X)):

# iterate through columns

for j in range(len(X[0])):

result[i][j] = X[i][j] + Y[i][j]

for r in result:

print(r)

**Class 14:**

**1.WAP to make use of exception handling using try except.**

import math

n=int(input("Enter any number" ))

try:

print(math.factorial(n))

except:

print("Enter only positive")

**2.WAP to make use of finally keywords.**

import math

n=int(input("Enter any number" ))

try:

print(math.factorial(n))

finally:

print("fgoodbyee")

print("Hello world")

**3.WAP to check string is pailaindrome or not.**

|  |  |
| --- | --- |
|  | wrd=input("Please enter a word") |
|  | wrd=str(wrd) |
|  | rvs=wrd[::-1] |
|  | print(rvs) |
|  | if wrd == rvs: |
|  | print("This word is a palindrome") |
|  | else: |
|  | print("This word is not a palindrome") |

**4.WAP in Python Program to count the number of lowercase letters and uppercase letters in a string.**

string=input("Enter string:")

count1=0

count2=0

**for**i**in**string:

**if**(i.islower()):

count1=count1+1

**elif**(i.isupper()):

count2=count2+1

**print**("The number of lowercase characters is:")

**print**(count1)

**print**("The number of uppercase characters is:")

**print**(count2)

**5.WAP in Python Program to find the binary equivalent of a number .**

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

a=[]

**while**(n>0):

dig=n%2

a.append(dig)

n=n//2

a.reverse()

**print**("Binary Equivalent is: ")

**for**i**in** a:

**print**(i,end=" ")

**6. Python Program to read a file and capitalize the first letter of every word in the file.**

fname = input("Enter file name: ")

with open(fname, 'r') as f:

for line in f:

l=line.title()

print(l)

**Class 15:**

**1.WAP to make a class and implement it.**

class A:

def getdata(self):

self.a=int(input("Enter marks of subjects"))

self.b=int(input("Enter marks of subjects"))

self.c=int(input("Enter marks of subjects"))

self.d=int(input("Enter marks of subjects"))

self.e=int(input("Enter marks of subjects"))

def putdata(self):

per=(self.a+self.b+self.c+self.d+self.e)/5

print(per)

obj=A()

obj.getdata()

obj.putdata()

**2.WAP to perform multilevel inheritance**

class A:

def hi(self):

print("A class is called")

class B(A):

def hello(self):

self.c=5

print("class B is called")

class C(B):

def bye(self):

self.c=10

print("class C is called")

obj=C()

obj.hello()

obj.bye()

obj.hi()

print(obj.c)

**3.WAP to perform hybrid inheritance**

class A:

def f(self):

print("class A")

class B:

def f1(self):

print("Class b")

class c(A,B):

def f2(self):

print("Class c")

class d(c):

def f3(self):

print("Class d")

obj=d()

obj.f()

obj.f1()

obj.f2()

obj.f3()

**4.WAP to implement polymorphism**

**classShark**():

**defswim**(self):

print("The shark is swimming.")

**defswim\_backwards**(self):

print("The shark cannot swim backwards, but can sink backwards.")

**defskeleton**(self):

print("The shark's skeleton is made of cartilage.")

**classClownfish**():

**defswim**(self):

print("The clownfish is swimming.")

**defswim\_backwards**(self):

print("The clownfish can swim backwards.")

**defskeleton**(self):

print("The clownfish's skeleton is made of bone.")

sammy = Shark()

sammy.skeleton()

casey = Clownfish()

casey.skeleton()

**5.WAP to implement constructor.**

class A:

def \_\_init\_\_(self,i,j):

self.b=i

self.h=j

def getdata(self):

print("your height width")

def putdata(self):

print(self.b,self.h)

i=int(input("Enter height"))

j=int(input("Enter width"))

obj=A(i,j)

obj.putdata()

obj.getdata()

obj.putdata()

**6.WAP to show polymorphism.**

class Parrot:

def \_\_init\_\_(self,i,j):

print(i+j)

def fly(self):

print("Parrot can fly")

def swim(self,a,b):

print("Parrot can't swim",a+b)

class Penguin(Parrot):

def fly(self):

super().\_\_init\_\_(20,20)

super().fly()

print("Penguin can't fly")

def swim(self):

super().swim(20,22)

print("Penguin can swim")

ob=Penguin(**10,20)**

**ob.fly()**

**ob.swim()**

**Class 16:**

**1.Write a Python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9)**

import re

defis\_allowed\_specific\_char(string):

charRe=re.compile(r'[^a-zA-Z0-9.]')

string =charRe.search(string)

returnnotbool(string)

print(is\_allowed\_specific\_char("ABCDEFabcdef123450"))

print(is\_allowed\_specific\_char("\*&%@#!}{"))

**2.Write a Python program that matches a string that has an *a* followed by zero or more b's.**

import re

deftext\_match(text):

patterns ='ab\*?'

ifre.search(patterns, text):

return'Found a match!'

else:

return('Not matched!')

print(text\_match("ac"))

print(text\_match("abc"))

print(text\_match("abbc"))

**3.Write a Python program that matches a string that has an *a* followed by three 'b'**

import re

deftext\_match(text):

patterns ='ab{3}?'

ifre.search(patterns, text):

return'Found a match!'

else:

return('Not matched!')

print(text\_match("abbb"))

print(text\_match("aabbbbbc"))

**4.Write a Python program to find sequences of lowercase letters joined with a underscore**

import re

deftext\_match(text):

patterns ='^[a-z]+\_[a-z]+$'

ifre.search(patterns, text):

return'Found a match!'

else:

return('Not matched!')

print(text\_match("aab\_cbbbc"))

print(text\_match("aab\_Abbbc"))

print(text\_match("Aaab\_abbbc"))

**5.Write a Python program to find sequences of one upper case letter followed by lower case letters.**

import re

deftext\_match(text):

patterns ='^[a-z]+\_[a-z]+$'

ifre.search(patterns, text):

return'Found a match!'

else:

return('Not matched!')

print(text\_match("aab\_cbbbc"))

print(text\_match("aab\_Abbbc"))

print(text\_match("Aaab\_abbbc"))

**6.Write a Python program that matches a string that has an 'a' followed by anything, ending in 'b'**

import re

deftext\_match(text):

patterns ='a.\*?b$'

ifre.search(patterns, text):

return'Found a match!'

else:

return('Not matched!')

print(text\_match("aabbbbd"))

print(text\_match("aabAbbbc"))

print(text\_match("accddbbjjjb"))

**7.Write a Python program that matches a word at the beginning of a string**

import re

deftext\_match(text):

patterns ='^\w+'

ifre.search(patterns, text):

return'Found a match!'

else:

return('Not matched!')

print(text\_match("The quick brown fox jumps over the lazy dog."))

print(text\_match(" The quick brown fox jumps over the lazy dog."))

**Class 17:**

**1.WAP to make server client connection**

**a.)server.py**

import socket

serversocket=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

port=9999

serversocket.bind(('127.0.0.1',port))

serversocket.listen(5)

clientsocket,addr=serversocket.accept()

print("got connection from %s" % str(addr))

msg="Thank tou for connecting"+"\r\n"

clientsocket.send(msg.encode("ascii"))

clientsocket.close()

**b)client.py**

import socket

s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

host=socket.gethostname()

port=9999

s.connect((host,port))

msg=s.recv(1025)

s.close()

print(msg.decode("ascii"))

**2.WAP to check if a number is an Armstrong number.**

n=int(input("Enter any number: "))

a=list(map(int,str(n)))

b=list(map(**lambda** x:x\*\*3,a))

**if**(sum(b)==n):

**print**("The number is an armstrong number. ")

**else**:

**print**("The number isn't an arsmtrong number. ")

**3.Find out even and odd numbers in a list.**

i=0

l1=[]

n=int(input("Enter how many number you want to enter"))

while i<n:

l1.append(int(input("Enter element of list")))

i=i+1

i=0

while i<n:

if l1[i]%2==0:

print("Even number:",l1[i])

else:

print("Odd number:",l1[i])

i=i+1

**4.WAP to search the elements in the list.**

i=0

l1=[10,20,30,40,50]

s=0

element=int(input("Enter any number you want to search"))

for j in l1:

if j==element:

s=s+1

new=int(input("Enter new number you want to replace"))

l1[i]=new

i=i+1

if s>=1:

print("Element found and replaced by new element list is:",l1)

else:

print("element not found list is:",l1)

**5.WAP to make pattern.**

**1**

**2 3**

**4 5 6**

**7 8 9 10**

i=0

j=0

s=0

while i<=4:

j=0

while j<i:

s=s+1

print(s,end=" ")

j=j+1

print("")

i=i+1

**6.WAPto search the number of times a particular number occurs in a list.**

a=[]

n=int(input("Enter number of elements:"))

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

b=int(input("Enter element:"))

a.append(b)

k=0

num=int(input("Enter the number to be counted:"))

**for** j **in** a:

**if**(j==num):

k=k+1

**print**("Number of times",num,"appearsis",k)

**7.WAP to implements multithreading.**

# Python program to illustrate the concept

# of threading

# importing the threading module

#from threading import \*

import threading

def print\_cube(num):

"""

function to print cube of given num

"""

print("Cube: {}".format(num \* num \* num))

def print\_square(num,L):

"""

function to print square of given num

"""

print(" Square: {}".format(num \* L))

def print\_powerfour(num):

'''

function print power four operation in the program

'''

print(" powerfour:{}".format(num\*num\*num\*num))

if \_\_name\_\_ == "\_\_main\_\_":

# creating thread

t1 = threading.Thread(target=print\_square, args=(10,3))

t2 = threading.Thread(target=print\_cube, args=(10,))

t3 = threading.Thread(target=print\_powerfour, args=(10,))

# starting thread 1

t1.start()

# starting thread 2

t2.start()

# starting thread 3

t3.start()

# wait until thread 1 is completely executed

t1.join()

# wait until thread 2 is completely executed

t2.join()

# wait until thread 3 is completely executed

t3.join()

# all the threads completely executed

print("Done!")

**class 18:**

**1.WAP to create a databse and table**

from sqlite3 import \*

con=connect("computec")

print("Database is created")

strcmd="create table employ(emp\_id varchar(20),emp\_name varchar(40),password varchar(15),confirm password varchar(15),address varchar(50))"

con.execute(strcmd)

print("Employ table is created")

con.close()

**2.WAP to enter values in table:**

import sqlite3

eid=123

ename=computec

esalary=9999

con=sqlite3.connect("uptec.db")

strcmd="insert into employ values("+eid+",'"+ename+"',"+esalary+")"

con.execute(strcmd)

print("Data is Sucessfully inserted")

con.commit()

con.close()

**3.WAP to see all the elements in database**

import sqlite3

con=sqlite3.connect("SIET.db")

sql="select \* from std"

cur=con.execute(sql)

r=cur.fetchall()

for i in r:

print(i)

con.close()

**4.WAP to update table.**

importsqlite3

# database name to be passed as parameter

conn =sqlite3.connect('mydatabase.db')

# update the student record

conn.execute("UPDATE Student SET name = 'Sam' where unix='B113059'")

conn.commit()

print("Total number of rows updated :", conn.total\_changes)

cursor =conn.execute("SELECT \* FROM Student")

forrow incursor:

   print (row)

conn.close()

**5.WAP to delete elements in table.**

# code for delete operation

import sqlite3

# database name to be passed as parameter

conn = sqlite3.connect('mydatabase.db')

# delete student record from database

conn.execute("DELETE from Student where unix='B113058'")

conn.commit()

print "Total number of rows deleted :", conn.total\_changes

cursor = conn.execute("SELECT \* FROM Student")

for row in cursor:

print row,

conn.close()

**6.WAP to create a database and input value from user .**

# code for executing query using input data

import sqlite3

# creates a database

con = sqlite3.connect("memory.db")

cur = con.cursor()

cur.execute("create table person (name, age, id)")

print ("Enter 5 students names:")

who = [input() for i in range(5)]

print ("Enter their ages respectively:")

age = [int(input()) for i in range(5)]

print ("Enter their ids respectively:")

p\_id = [int(input()) for i in range(5)]

n = len(who)

for i in range(n):

# This is the q-mark style:

cur.execute("insert into person values (?, ?, ?)", (who[i], age[i], p\_id[i]))

# And this is the named style:

cur.execute("select \* from person")

# Fetches all entries from table

print (cur.fetchall())

**Class 19**

**1.WAP to make implementation of labels and buttons in python.**

from tkinter import \*

frm=Tk()

txt=StringVar()

lab=Label(text="Enter Text:")

lab.grid(row=0,column=0)

u=Entry(frm,textvariable=txt)

u.grid(row=0,column=1)

c1=Checkbutton(frm,text="male",variable="g1")

c2=Checkbutton(frm,text="female",variable="g2")

c1.place(x=10,y=10)

c2.place(x=40,y=50)

r1=Radiobutton(frm,text="male",variable="gender",value="male")

r2=Radiobutton(frm,text="female",variable="gender",value="female")

r3=Radiobutton(frm,text="other",variable="gender",value="other")

r1.place(x=10,y=10)

r2.place(x=40,y=50)

r3.place(x=60,y=20)

btn=Button(frm,text="copy",command=value)

btn.grid(row=3,column=0,padx=5,pady=20)

frm.mainloop()

**2.WAP to concatenate first name and last name in gui using another entry and function.**

from tkinter import \*

frm=Tk()

txt=StringVar()

txt2=StringVar()

txt3=StringVar()

lab=Label(text="first name:")

lab.grid(row=0,column=0)

u=Entry(frm,textvariable=txt)

u.grid(row=0,column=1)

lab2=Label(text="last name:")

lab2.grid(row=1,column=0,pady=20)

v=Entry(frm,textvariable=txt2)

v.grid(row=1,column=1,pady=20)

lab3=Label(text="Full Name:")

lab3.grid(row=2,column=0,pady=20)

w=Entry(frm,textvariable=txt3)

w.grid(row=2,column=1,pady=20)

def fun():

fname=u.get()

lname=v.get()

txt3.set(fname+" "+lname)

def fun1():

txt.set("")

txt2.set("")

txt3.set("")

btn=Button(frm,text="concatenate",command=fun)

btn.grid(row=3,column=0,padx=5,pady=20)

btn1=Button(frm,text="Cancel",command=fun1)

btn1.grid(row=3,column=1,padx=8,pady=20)

frm.mainloop()

**3.WAP to copy text from one entry to other.**

from tkinter import \*

frm=Tk()

txt=StringVar()

txt1=StringVar()

lab=Label(text="Enter Text:")

lab.grid(row=0,column=0)

u=Entry(frm,textvariable=txt)

u.grid(row=0,column=1)

def value():

name=u.get()

txt1.set(name)

def dele():

txt.set("")

txt1.set("")

lab2=Label(text="Copied Text:")

lab2.grid(row=1,column=0,pady=20)

v=Entry(frm,textvariable=txt1)

v.grid(row=1,column=1,pady=20)

btn=Button(frm,text="copy",command=value)

btn.grid(row=3,column=0,padx=5,pady=20)

btn1=Button(frm,text="Cancel",command=dele)

btn1.grid(row=3,column=1,padx=8,pady=20)

frm.mainloop()

**4.WAP to create frame**

from tkinter import \*

root = Tk()

frame = Frame(root,bd=2)

frame.pack()

bottomframe = Frame(root,bd=5)

bottomframe.pack( side = BOTTOM )

redbutton = Button(frame, text="Red", fg="red")

redbutton.pack( side = LEFT)

greenbutton = Button(frame, text="Brown", fg="brown")

greenbutton.pack( side = LEFT )

bluebutton = Button(frame, text="Blue", fg="blue")

bluebutton.pack( side = LEFT )

blackbutton = Button(bottomframe, text="Black", fg="black")

blackbutton.pack( side = BOTTOM)

root.mainloop()

**5.WAP to implement radio button.**

from tkinter import \*

frm=Tk()

r1=Radiobutton(frm,text="male",variable="gender",value="male")

r2=Radiobutton(frm,text="female",variable="gender",value="female")

r3=Radiobutton(frm,text="other",variable="gender",value="others")

r1.place(x=10,y=10)

r2.place(x=40,y=50)

r3.place(x=60,y=20)

frm.mainloop()

**6.WAP to implement check box.**

from tkinter import \*

frm=Tk()

c1=Checkbutton(frm,text="male",variable="g1")

c2=Checkbutton(frm,text="female",variable="g2")

c1.place(x=10,y=10)

c2.place(x=40,y=50)

**7.WAP to make spinbox.**

from tkinter import \*

master = Tk()

w = Spinbox(master, from\_=0, to=10)

w.pack()

mainloop()

**8.WAP to insert image**

from tkinter import \*

window=Tk()

photo=PhotoImage(file="a.jpg")

l=l(root,image=photo)

l.pack

window.mainloop()

**9.WAP to create login form.**

from tkinter import \*

frm=Tk()

frm.geometry('400x400')

r=Label(frm,text="Login Form",font=("MV Boli",25))

r.place(x=100,y=0)

a=Label(frm,text="Username")

b=Label(frm,text="Password")

c=Label(frm,text="Confirm Password")

a.place(x=85,y=50)

b.place(x=85,y=80)

c.place(x=40,y=110)

ent1=Entry(frm)

ent2=Entry(frm,show='\*')

ent3=Entry(frm,show='\*')

ent1.place(x=150,y=50)

ent2.place(x=150,y=80)

ent3.place(x=150,y=110)

def valid():

t2=ent2.get()

t3=ent3.get()

if(t3==t2):

print("Log in success")

else:

print("Keep trying")

b1=Button(frm,text="Log-in",bg="skyblue",command=valid)

b1.place(x=120,y=150)

b2=Button(frm,text="Cancel",bg="skyblue",command=reg)

b2.place(x=200,y=150)

frm.mainloop()

**10.WAP to make Regestration form.**

from tkinter import \*

from sqlite3 import \*

sr=Tk()

sr.geometry('500x500')

sr.title("COMPUTECH REGISTRATION FORM")

sr.configure(background="grey")

def save():

print("data is saved")

def cancel():

sr.destroy()

Lab1=Label(sr,text="STUDENT REGISTRATION FORM",bg="grey",font=('timesnewroman 20 bold underline'))

Lab1.place(x=400,y=50)

Lab2=Label(sr,text="First Name:",bg='grey',font=('timesnewroman 12'))

Lab2.place(x=80,y=100)

v=Entry(sr)

v.place(x=170,y=100)

Lab2=Label(sr,text="(max 30 characters a-z & A-Z)",bg="grey")

Lab2.place(x=300,y=100)

Lab3=Label(sr,text="Last Name:",bg='grey',font=('timesnewroman 12'))

Lab3.place(x=80,y=140)

w=Entry(sr)

w.place(x=170,y=140)

Lab4=Label(sr,text="Date Of Birth:`",bg='grey',font=('timesnewroman 12'))

Lab4.place(x=80,y=180)

d=StringVar()

l=[1,2,3,4,5,6,7,8,9,10,11,12,13,14]

opt=OptionMenu(sr,d,\*l)

opt.place(x=190,y=180)

Lab5=Label(sr,text="Email id:",bg='grey',font=('timesnewroman 12'))

Lab5.place(x=80,y=220)

y=Entry(sr)

y.place(x=170,y=220)

Lab6=Label(sr,text="Mobile no.",bg='grey',font=('timesnewroman 12'))

Lab6.place(x=80,y=260)

z=Entry(sr)

z.place(x=170,y=260)

lab7=Label(sr,text="Gender",bg='grey',font=('timesnewroman 12'))

lab7.place(x=80,y=300)

rb=StringVar()

btn1=Radiobutton(sr,text="Male",variable=rb,value='male',bg='grey',font=('timesnewroman 12'))

btn1.place(x=170,y=300)

btn2=Radiobutton(sr,text="Female",variable=rb,value='female',bg='grey',font=('timesnewroman 12'))

btn2.place(x=250,y=300)

lab8=Label(sr,text="Adress",bg='grey',font=('timesnewroman 12'))

lab8.place(x=80,y=340)

a=Entry(sr)

a.place(x=170,y=340)

lab9=Label(sr,text="City",bg='grey',font=('timesnewroman 12'))

lab9.place(x=80,y=410)

b=Entry(sr)

b.place(x=170,y=410)

lab10=Label(sr,text="Pin Code",bg='grey',font=('timesnewroman 12'))

lab10.place(x=80,y=450)

c=Entry(sr)

c.place(x=170,y=450)

lab11=Label(sr,text="State",bg='grey',font=('timesnewroman 12'))

lab11.place(x=80,y=490)

d=Entry(sr)

d.place(x=170,y=490)

lab12=Label(sr,text="Country",bg='grey',font=('timesnewroman 12'))

lab12.place(x=80,y=530)

e=Entry(sr)

e.place(x=170,y=530)

button1=Button(sr,text="save",bg="grey",bd=5,width=4,command=save)

button1.place(x=80,y=580)

button2=Button(sr,text="cancel",bg="grey",bd=5,width=4,command=cancel)

button2.place(x=180,y=580)

sr.mainloop()

**11.WAP to create a panel for user of employee in which save data and show previous data**

from tkinter import \*

frm=Tk()

txt=StringVar()

txt2=StringVar()

txt3=StringVar()

lab=Label(text="Employee id:")

lab.grid(row=0,column=0)

u=Entry(frm,textvariable=txt)

u.grid(row=0,column=1)

lab2=Label(text="Employee name:")

lab2.grid(row=1,column=0,pady=20)

v=Entry(frm,textvariable=txt2)

v.grid(row=1,column=1,pady=20)

lab3=Label(text="Salary:")

lab3.grid(row=2,column=0,pady=20)

w=Entry(frm,textvariable=txt3)

w.grid(row=2,column=1,pady=20)

def call():

import sqlite3

eid=u.get()

ename=v.get()

esalary=w.get()

con=sqlite3.connect("SIET.db")

strcmd="insert into emp values("+eid+",'"+ename+"',"+esalary+")"

con.execute(strcmd)

lab4=Label(text="Data is Sucessfully inserted")

lab4.grid(row=3,column=2,pady=50)

con.commit()

con.close()

def show():

import sqlite3

con=sqlite3.connect("SIET.db")

strcmd="select \* from emp"

cur=con.execute(strcmd)

r=cur.fetchall()

for i in r:

print(i)

con.close()

btn=Button(frm,text="Save",command=call)

btn.grid()

btn=Button(frm,text="Show",command=show)

btn.grid()

frm.mainloop()