CYCLE-1

1. **WRITE A PYTHON PROGRAM TO FIND THE AREA OF A CIRCLE**

**AIM** :

Area of a circle.

**ALGORITHM :**

STEP 1: START

STEP 2: import pi from math

STEP 3: Read r, radius

STEP 4: Apply pi\*r\*r

STEP 5: STOP.

**SOURCE CODE :**

from math import pi

r = float(input ("Input the radius of the circle : "))

print ("The area of the circle with radius " + str(r) + " is: " + str(pi \* r\*\*2))

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO FIND OUT SIMPLE INTEREST**

**AIM :**

Program to find simple interest.

**ALGORITHM :**

STEP 1: START

STEP 2: Define a function simple\_interest(p, t, r)

STEP 2: Read pr, ti, ra (principal, time period, rate of interest)

STEP 3: When the function is called, perform (p\*t\*r)/100

STEP 4: The result is returned.

STEP 5: STOP.

**SOURCE CODE:**

def simple\_interest(p,t,r):

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

print('The Simple Interest is', si)

return si

pr=float(input("Enter the principle: "))

ti=float(input("Enter the time: "))

ra=float(input("Enter the rate of interest: "))

simple\_interest(pr,ti,ra)

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PROGRAM TO SWAP TWO NUMBERS WITHOUT USING TEMPORARY VARIABLES.**

**AIM :**

Program to swap two numbers without using temporary variables.

**ALGORITHM :**

STEP 1: START

STEP 2: Read x and y

STEP 3: Print the values before swapping

STEP 4: Swap the values – x,y=y,x

STEP 5: Print the values after swapping

STEP 6: STOP.

**SOURCE CODE :**

x = int(input("Enter the first value: "))

y = int(input("Enter the second value: "))

print ("Before swapping: ")

print("Value of x : ", x, " and y : ", y)

x, y = y, x

print ("After swapping: ")

print("Value of x : ", x, " and y : ", y)

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO CONVERT TEMPERATURE FROM CELSIUS TO FAHRENHEIT.**

**AIM :**

Convert temperature from Degree Celsius to Fahrenheit.

**ALGORITHM :**

STEP 1: START

STEP 2: Read the temperature in Degree Celsius

STEP 3: Apply (d\*1.8)+32

STEP 4: Print the converted temperature.

STEP 5: STOP.

**SOURCE CODE :**

d=float(input("Temperature in Degree Celsius: "))

f=(d\*1.8)+32

print("Converted temperature is: ",f,"F")

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO CHECK WHETHER A GIVEN NUMBER IS ODD OR EVEN.**

**AIM :**

Program to check whether a given number is odd or even.

**ALGORITHM :**

STEP 1: START

STEP 2: Read a number

STEP 3: Using if check whether the number is divisible by 2- if(num%2)==0

STEP 4: Print the result

STEP 5: STOP.

**SOURCE CODE :**

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

if (num % 2) == 0:

print(str(num) + " is an even number")

else:

print(str(num) + " is an odd number")

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO FIND THE LARGEST OF THREE NUMBERS.**

**AIM :**

Program to find the largest of three numbers.

**ALGORITHM :**

STEP 1: START

STEP 2: Read a, b, c

STEP 3: Compare the numbers using if..elif..else

STEP 4: Print the result.

STEP 5: STOP.

**SOURCE CODE :**

a=int(input("Enter the first number: "))

b=int(input("Enter the second number: "))

c=int(input("Enter the third number: "))

if a>b and a>c:

print(a,"is the largest number")

elif b>c:

print(b,"is the largest number")

else:

print(c,"is the largest number")

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO CHECK WHETHER A GIVEN YEAR IS LEAP YEAR OR NOT.**

**AIM :**

Check whether the given year is leap year or not.

**ALGORITHM :**

STEP 1: Read the year.

STEP 2: Using nested if check if the year is leap year or not.

STEP 3: Print the result.

STEP 4: STOP.

**PROGRAM**

year = int(input("Enter the year: "))

if (year % 4) == 0:

if (year % 100) == 0:

if (year % 400) == 0:

print(str(year) + " is a leap year")

else:

print(str(year) + " is not a leap year")

else:

print(str(year) + " is a leap year")

else:

print(str(year) + " is not a leap year")

**RESULT :**

Output is obtained and the result is verified.

CYCLE-2

1. **WRITE A PYTHON PROGRAM TO GENERATE A LIST OF POSITIVE NUMBERS FROM A GIVEN LIST OF INTEGERS.**

**AIM** :

Program to generate a list of positive numbers from a given list of integers.

**ALGORITHM :**

STEP 1: Read N, the total number of elements in the list.

STEP 2: Enter the elements and append it in lists

STEP 3: Print the list of elements that are greater than 0

STEP 4: STOP.

**SOURCE CODE :**

N=int(input("Enter Total number of elements in list : "))

lists=[]

for i in range(N):

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

lists.append(value)

test = [each for each in lists if each>0]

print(test)

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO FORM A LIST OF VOWELS SELECTED FROM A GIVEN WORD USING LIST COMPREHENSION.**

**AIM :**

PROGRAM TO FORM A LIST OF VOWELS SELECTED FROM A WORD USING LIST COMPREHENSION.

**ALGORITHM :**

STEP 1: START

STEP 2: Read a word

STEP 3: Find the vowels from the word and append in list[]

STEP 4: Print the result

STEP 5: STOP

**SOURCE CODE :**

a= input("ENTER THE WORD: ")

list=[]

vowels="AaEeIiOoUu"

ans=set(each for each in a if each in vowels)

list.append(ans)

print(list)

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO GET A STRING FROM THE GIVEN STRING WHERE ALL OCCURRENCES OF ITS FIRST CHAR HAVE BEEN CHANGED TO ‘$’ EXCEPT THE FIRST CHAR ITSELF.**

**AIM :**

PROGRAM TO GET A STRING FROM THE GIVEN STRING WHERE ALL OCCURRENCES OF ITS FIRST CHAR HAVE BEEN CHANGED TO ‘$’ EXCEPT THE FIRST CHAR ITSELF.

**ALGORITHM :**

STEP 1: START

STEP 2: Define a function lis[]

STEP 3: Perform lis.append(‘$’) and lis.append(ch)

STEP 4: Print the result

STEP 5: STOP

**SOURCE CODE :**

lis []

s=input(“Enter string”)

for ch in s:

if ch in lis:

lis.append(‘$’)

else:

lis.append (ch)

print(”.join(str(i) for i in lis))

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO COUNT THE OCCURRENCES OF EACH WORD IN A LINE OF TEXT.**

**AIM :**

PROGRAM TO COUNT THE OCCURRENCES OF EACH WORD IN A LINE OF TEXT.

**ALGORITHM :**

STEP 1: START

STEP 2: Take user input of string and substring

STEP 3: Find the result by performing count = string.count(substring)

STEP 4: Print the result

STEP 5: STOP

**SOURCE CODE :**

string = input(“Enter a string: “)

substring = input(“Enter a substring: “)

count = string.count(substring)

print(“The count is: ”, count)

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM THAT SORTS DICTIONARIES IN ASCENDING AND DESCENDING ORDER.**

**AIM :**

PROGRAM THAT SORTS DICTIONARIES IN ASCENDING AND DESCENDING ORDER.

**ALGORITHM :**

STEP 1: START

STEP 2:Create dic{}

STEP 3: Get user inputs n1 and dic[i]

STEP 4: Print(sorted(dic.items(),key = lambda kv:(kv[1],kv[0])))

STEP 5: STOP.

**SOURCE CODE :**

dic={}

n1=int(input("Total number of elements in dict 1 : "))

for i in range(n1):

dic[i]=input("Enter element : ")

print(sorted(dic.items(), key = lambda kv:(kv[1], kv[0])))

print(sorted(dic.items(), key = lambda kv:(kv[1], kv[0]), reverse=True))

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO MERGE TWO DICTIONARIES.**

**AIM :**

PROGRAM TO MERGE TWO DICTIONARIES.

**ALGORITHM :**

STEP 1: START

STEP 2: Create three dictionaries dict1 ,dict2 and dict3

STEP 3: Give dict1 for i in range(n1) and dict2 for i in range(3,3+n2)

STEP 4: Print dict3

STEP 5: STOP.

**SOURCE CODE :**

dict1={}

dict2={}

dict3={}

n1=int(input(“Total number of elements in dict 1: “))

for i in range(n1):

dict1[i]=int(input(“Enter element: “))

n2=int(input(“Total number of elements in dict 2: “))

for i in range(3,3+n2):

dict2[i]=int(input(“Enter element: “))

print(dict1)

print(dict2)

for i in dict1:

if i not in dict2:

new = {i:dict[i]}

dict3.update(new)

for in in dict2:

if i not in dict1:

new = {i:dict2[i]}

dict3.update(new)

print(dict3)

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO FIND GCD OF 2 NUMBERS.**

**AIM :**

PROGRAM TO FIND GCD OF 2 NUMBERS.

**ALGORITHM :**

1. START

2.Define def find\_hcf(a,b) while(b):

a,a = b, a % b

3.Enter user inputs a and b

4. find hcf using num = find\_hcf(a,b)

5.Print H.C.F

6.STOP

**SOURCE CODE :**

def find\_hcf(a, b):

while(b):

a, a = b, a % b

return a

a = int(input("Enter the first number: "))

b = int(input("Enter the second number: "))

num = find\_hcf(a,b)

print("The H.C.F. is: ")

print(num)

**RESULT :**

Output is obtained and the result is verified.

CYCLE-3

1. **WRITE A PYTHON PROGRAM TO FIND THE FACTORIAL OF A NUMBER USING FUNCTIONS**

**AIM :**

PROGRAM TO FIND THE FACTORIAL OF A NUMBER USING FUNCTIONS.

**ALGORITHM :**

STEP 1: START

STEP 2: Define a function fact(n)

STEP 3: Read num

STEP 4: When the function is called, the factorial is computed by performing n\*fact(n-1) and the value is returned, which is then displayed.

Step 5: STOP.

**SOURCE CODE :**

def fact(n):

if n == 1:

return n

else:

return n\*fact(n-1)

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

if num < 0:

print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

print("The factorial of",num,"is", fact(num))

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PROGRAM TO GENERATE FIBONACCI SERIES OF N TERMS USING FUNCTIONS.**

**AIM :**

PROGRAM TO GENERATE FIBONACCI SERIES OF N TERMS USING FUNCTIONS.

**ALGORITHM :**

STEP 1: START

STEP 2: Define a function fib(n)

STEP 3: Read nterms

STEP 4: When the function fib() is called, the Fibonacci sequence of numbers upto n terms is found by performing-

(fib(n-1) + fib(n-2)) , and the result is returned and displayed.

**SOURCE CODE :**

def fib(n):

if n <= 1:

return n

else:

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

nterms = int(input("Enter the limit: "))

if nterms <= 0:

print("Plese enter a positive integer")

else:

print("Fibonacci sequence:")

for i in range(nterms):

print(fib(i))

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO DISPLAY THE GIVEN PYRAMID WITH THE STEP NUMBER ACCEPTED FROM USER USING FUNCTIONS.**

**Eg: N=4**

**1**

**2 4**

**3 6 9**

**4 8 12 16**

**AIM :**

PROGRAM TO DISPLAY THE GIVEN PYRAMID WITH THE STEP NUMBER ACCEPTED FROM USER USING FUNCTIONS.

**ALGORITHM :**

STEP 1: START

STEP 2: Read rows

STEP 3: Using nested for loop, increment the values and perform i \* j , and finally print the pattern upto the user input N

STEP 4: STOP.

**SOURCE CODE :**

rows = int(input("Enter the number of rows, N = "))

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

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

square = i \* j

print(i \* j, end=")

print()

RESULT :

Output is obtained and the result is verified

1. **WRITE A LAMBDA FUNCTIONS TO FIND THE AREA OF SQUARE, RECTANGLE AND TRIANGLE.**

**AIM :**

PROGRAM TO WRITE LAMBDA FUNCTIONS TO FIND THE AREA OF SQUARE, RECTANGLE AND TRIANGLE.

**ALGORITHM :**

STEP 1: START

STEP 2: area\_of\_a\_square is a function that takes two integers and return a\*a

STEP 3: area\_of\_a\_rectangle is a function that takes an integer and return a\*b

STEP 4: area\_ of\_a\_triangle is a function that takes two integers and return (a+b+c)/2

STEP 5: Returns and prints the result corresponding to the values

STEP 6: STOP.

**SOURCE CODE :**

area\_square = lambda side : side \* side

area\_rectangle = lambda length,width : length \* width

area\_triangle = lambda s,a,b,c : (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

a=10

b=20

c=15

s = (a + b + c) / 2

print(area\_square(a))

print(area\_rectangle(a,b))

print(area\_triangle(s,a,b,c))

**RESULT :**

Output is obtained and the result is verified.

1. **CREATE A PACKAGE GRAPHICS WITH MODULES RECTANGLE, CIRCLE AND SUB-PACKAGE 3D GRAPHICS WITH MODULES CUBOID AND SPHERE.INCLUDE METHODS TO FIND AREA AND PERIMETER OF RESPECTIVE FIGURES IN EACH MODULE.WRITE PROGRAMS THAT FIND AREA AND PERIMETER OF FIGURES BY DIFFERENT IMPORTING STATEMENTS.**

**ALGORITHM :**

STEP 1: START

STEP 2: Create a folder graphics and in it store: rectangle.py, circle.py and init.py

STEP 3: Create another folder in graphics and in it store: cuboid.py, sphere.py and init.py

STEP 4: Import these packages into another program and print the output

STEP 5: STOP

**SOURCE CODE :**

import graphics

from graphics import circle,rectangle

from graphics.tdgraphics import cuboid,sphere

from graphics.circle import \*

print("Area of a circle with radius 10 is : ",circle.area\_circle(10))

print("Permeter of a circle with radius 10 is ",circle.perimeter\_circle(10))

print("Area of a circle with radius 10 is : ",area\_circle(10))

print("Area of a Rectangle with length and width 10 is : ",rectangle.area\_rec(10,10))

print("Permeter of a Rectangle with length and width 10 is : ",rectangle.perimeter\_rec(10,10))

print("Area of a cuboid with length,width,height 10 is : ",cuboid.area\_cuboid(10,10,10))

print("Permeter of a cuboid with length,width,height 10 is : ",cuboid.perimeter\_cuboid(10,10,10))

print("Area of a spere with radius 10 is : ",sphere.area\_sphere(10))

print("Permeter of a spere with radius 10 is ",sphere.perimeter\_sphere(10))

**//Graphics:**

rectangle.py

def area\_rec(length,width):

return length\*width

def perimeter\_rec(length,width):

return 2\*(length+width)

circle.py

from math import pi

def area\_circle(radius):

return pi\*radius\*radius

def perimeter\_circle(radius):

return 2\*pi\*radius

cuboid.py

def area\_cuboid(l,b,h):

return 2\*(l\*h + b\*h + l\*b)

def perimeter\_cuboid(l,b,h):

return 4\*(l+b+h)

sphere.py

from math import pi

def area\_sphere(radius):

return 4\*(pi\*radius\*radius)

def perimeter\_sphere(radius):

return 2\*pi\*radius

**RESULT :**

Output is obtained and the result is verified.

1. **CREATE A BANK ACCOUNT WITH MEMBERS ACCOUNT NUMBER, NAME, TYPE OF ACCOUNT AND BALANCE. WRITE CONSTRUCTOR AND METHODS TO DEPOSIT AT THE BANK AND WITHDRAW AN AMOUNT FROM THE BANK.**

**AIM :**

Write a program to create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

**ALGORITHM :**

STEP 1: START

STEP 2: Read input as name, account number, type, balance

STEP 3: Call function as per the entered option

STEP 4: View details as per the predefined function.

STEP 5: STOP.

**SOURCE CODE** :

class bank:

\_\_acc\_name=""

\_\_acc\_no = ""

\_\_acc\_type = ""

\_\_acc\_balance = 0

def \_\_init\_\_(self,a\_name,a\_no,a\_type,a\_balance):

self.\_\_acc\_name = a\_name

self.\_\_acc\_no = a\_no

self.\_\_acc\_type = a\_type

self.\_\_acc\_balance = a\_balance

def deposit(self,a\_deposit):

print("Initial balance is : ",self.\_\_acc\_balance)

self.\_\_acc\_balance += a\_deposit

print("Current balance is : ",self.\_\_acc\_balance)

def withdraw(self):

print("Current balance is : ",self.\_\_acc\_balance)

self.amount = int(input("Please enter the amount you want to withdraw: "))

if self.amount > self.\_\_acc\_balance:

print("You don't have enough balance to withdraw ")

print("Current balance is : ",self.\_\_acc\_balance)

else:

print(self.amount," has been withdrawn .")

self.\_\_acc\_balance -= self.amount

print("Current balance is : ",self.\_\_acc\_balance)

def acc\_info(self):

print("Account holder name : ",self.\_\_acc\_name)

print("Account number : ",self.\_\_acc\_no)

print("Account type : ",self.\_\_acc\_type)

print("Account Balance is : ",self.\_\_acc\_balance)

def main():

name = input("Enter Account holder name : ")

no = input("Enter Account number : ")

atype = input("Enter Account type : ")

bal = int(input("Enter Account initial balance : "))

holder = bank(name,no,atype,bal)

while(True):

print("\n")

opt = int(input("1)Deposit \n2)Withdraw \n3)Account info \n0)Exit\nChoose your option :: "))

if opt == 1:

amount = int(input("Deposit amount : "))

holder.deposit(amount)

elif opt == 2:

holder.withdraw()

elif opt == 3:

holder.acc\_info()

elif opt == 0:

break

else:

print("Invalid Option !")

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

while(True):

main()

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PROGRAM TO CREATE A CLASS RECTANGLE WITH PRIVATE ATTRIBUTES LENGTH AND WIDTH. OVERLOAD ‘<’ OPERATOR TO COMPARE THE AREA OF 2 RECTANGLES.**

**AIM :**

Write a program to create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.

**ALGORITHM :**

STEP 1: START

STEP 2: Take input as length and breadth from two rectangles

STEP 3: Calculate area of rectangles and compare them.

STEP 4: Print the result of area and the largest rectangle among them.

STEP 5: STOP

**SOURCE CODE :**

class rectangle:

\_\_area = 0

\_\_perimeter = 0

def \_\_init\_\_(self,length,width):

self.\_\_length = length

self.\_\_width = width

def calc\_area(self):

self.\_\_area = self.\_\_length\*self.\_\_width

print("Area is :",self.\_\_area)

def \_\_lt\_\_(self,second):

if self.\_\_area < second.\_\_area:

return True

else:

return False

length1= int(input("Enter length of the rectangle 1 : "))

width1 = int(input("Enter breadth of the rectangle 1 : "))

length2 = int(input("Enter length of the rectangle 2 : "))

width2 = int(input("Enter breadth of the rectangle 2 : "))

obj1 = rectangle(length1,width1)

obj2 = rectangle(length2,width2)

obj1.calc\_area()

obj2.calc\_area()

if obj1 < obj2:

print("Rectangle two is large")

elif obj1>obj2:

print("Rectangle one is large ")

else:

print("These are equal")

**RESULT :**

Output is obtained and the result is verified.

1. **WRITE A PYTHON PROGRAM TO READ NUMBERS FROM A FILE AND STORE ODD AND EVEN NUMBERS SEPERATELY IN TWO DIFFERENT FILES.**

**AIM:**

Write a python program to read numbers from a file and store odd and even numbers seperately in two different files

**ALGORITHM:**

STEP 1: START

STEP 2: Create 3 files- file1.txt, oddfile.txt, evenfile.txt

STEP 3: Open files, f= open("file1.txt","r") , f1= open("oddfile.txt","w") , f2=open("evenfile.txt","w")

STEP 4: Write the odd numbers from f into f1 and even numbers from f into f2

STEP 5: Close f, f1, f2

STEP 6: STOP.

**SOURCE CODE:**

f=open("file1.txt","r")

f1=open("oddfile.txt","w")

f1=open("oddfile.txt","w")

for i in map(int,f.readline().split( )):

if i%2!=0:

f1.write(str(i)+' ')

else:

f2.write(str(i)+' ')

f.close()

f1.close()

f2.close()

**RESULT:**

Output is obtained and the result is verified.