**Experiment 13**

**Aim:** Write a python program to implement multithreaded applications in python.

**Program:**

**Thread.py**

import thread # import the thread module

import time # import time module

def cal\_sqre(num): # define the cal\_sqre function

print(" Calculate the square root of the given number")

for n in num:

time.sleep(0.3) # at each iteration it waits for 0.3 time

print(' Square is : ', n \* n)

def cal\_cube(num): # define the cal\_cube() function

print(" Calculate the cube of the given number")

for n in num:

time.sleep(0.3) # at each iteration it waits for 0.3 time

print(" Cube is : ", n \* n \*n)

arr = [9, 12, 8, 6, 4] # array

t1 = time.time() # get total time to execute the functions

cal\_sqre(arr) # call cal\_sqre() function

cal\_cube(arr) # call cal\_cube() function

print(" Total time taken by threads is :", time.time() - t1) # print the total time

**Output:**

D:\Python Programs>python Thread.py

Calculate the square root of the given number

Square is : 81

Square is : 144

Square is : 64

Square is : 36

Square is : 16

Calculate the cube of the given number

Cube is : 729

Cube is : 1728

Cube is : 512

Cube is : 216

Cube is : 64

Total time taken by threads is : 3.0044047832489014

**Synchronizing Threads in Python:**

**Thread.py**

import time # import time module

import threading

from threading import \*

def cal\_sqre(num): # define a square calculating function

print(" Calculate the square root of the given number")

for n in num: # Use for loop

time.sleep(0.3) # at each iteration it waits for 0.3 time

print(' Square is : ', n \* n)

def cal\_cube(num): # define a cube calculating function

print(" Calculate the cube of the given number")

for n in num: # for loop

time.sleep(0.3) # at each iteration it waits for 0.3 time

print(" Cube is : ", n \* n \*n)

ar = [9, 12, 8, 6, 4] # array

t = time.time() # get total time to execute the functions

#cal\_cube(ar)

#cal\_sqre(ar)

th1 = threading.Thread(target=cal\_sqre, args=(ar, ))

th2 = threading.Thread(target=cal\_cube, args=(ar, ))

th1.start()

th2.start()

th1.join()

th2.join()

print(" Total time taking by threads is :", time.time() - t) # print the total time

print(" Again executing the main thread")

print(" Thread 1 and Thread 2 have finished their execution.")

**Output:**

D:\Python Programs>python Threaded.py

Calculate the square root of the given number

Calculate the cube of the given number

Square is : 81

Cube is : 729

Square is : 144

Cube is : 1728

Square is : 64

Cube is : 512

Square is : 36

Cube is : 216

Square is : 16

Cube is : 64

Total time taking by threads is : 1.502774715423584

Again executing the main thread

Thread 1 and Thread 2 have finished their execution.

**Conclusion:** Hence we have implemented multithreaded applications in python.