# Python program to illustrate the concept

# of threading

# importing the threading module

import threading

def print\_cube(num):

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

def print\_square(num):

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

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

# creating thread

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

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

# starting thread 1

t1.start()

# starting thread 2

t2.start()

# wait until thread 1 is completely executed

t1.join()

# wait until thread 2 is completely executed

t2.join()

# both threads completely executed

print("Done!")

Program 2 :

import threading

import os

def task1():

print("Task 1 assigned to thread: {}".format(threading.current\_thread().name))

print("ID of process running task 1: {}".format(os.getpid()))

def task2():

print("Task 2 assigned to thread: {}".format(threading.current\_thread().name))

print("ID of process running task 2: {}".format(os.getpid()))

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

# print ID of current process

print("ID of process running main program: {}".format(os.getpid()))

# print name of main thread

print("Main thread name: {}".format(threading.current\_thread().name))

# creating threads

t1 = threading.Thread(target=task1, name='t1')

t2 = threading.Thread(target=task2, name='t2')

# starting threads

t1.start()

t2.start()

# wait until all threads finish

t1.join()

t2.join()