24/6

Multi-Tasking :-

There are 2 types of Multi-Tasking.

(i) Thread-Based Multitasking
(ii) Process-Based Multitasking

→ Process-Based Multi-tasking -

- they can do different process based on CPU, GPU so on.

→ Thread-Based Multitasking -

* It can also known as Multithreading.

By default in python, multithreading is not supported.

In python, there is threading module to support multithreading applications these threading modules are used.

```
import threading
print("current thread", threading.current thread().getName())
```

O/p: Current thread MainThread

By default there is a current thread namely main thread.

Note:- Whatever we create after start it treats as main thread.

```
from threading import *
def display():
    for i in range(10):
        print("child thread")
t=Thread(target=display)
t.start()
```

```
for i in range(10):
    print("main thread")
```

o/p: main thread child thread

:

main thread child thread (10 times)

Let to represent this in Object Oriented programming, we can also use class as follows.

```
from threading import *
class Mythread(Thread):
    def run(self):
        for i in range(10):
            print("child thread")
t=Mythread()
t.start()
for i in range(10):
    print("main thread")
```

o/p: Child thread main thread

:

child thread main thread (loop for 10 times)

Note: The range of both for loops should not only same, they need to

Let now. See the Extending thread

```
from threading import *
class Mythread:
    def display(self):
        for i in range(10):
            print("child thread") :

t=MyThread()
obj=Thread(target=t.display())
obj.start()
for i in range(10):
    print ("main thread")
```

o/p: child thread

:

child thread (10 times)
main thread

:

main thread (10 times)

As it will not Execute Simultaneously, since it means Somewhere the code has been wrong.

```
from threading import *
class Mythread:
    def display(self):
        for i in range(10):
            print("child thread")
t=Mythread()
obj=Thread(target=t.display)
obj.start()
```

```
for i in range(10):
    print("main thread")
```

o/p:- Child threadmain thread

:

Child threadmain thread. (10 times)

* Without Multi-threading:-

```
import time
def double(numbers):
    for i in numbers:
        time.sleep(1)
        print("double of that number", i*2)
def triple(numbers):
    for i in numbers:
        time.sleep(1)
        print("triple of a number", 3*i)
numbers = [1,2,3,4,5,6]
begintime = time.time()
double(numbers)
triple(numbers)
print("total time taken", time.time()-beginnertime)
```

o/p:- double of that number 2

double of that number 4

double of that number 6

double of that number 8

double of that number 10
double of that number 12
triple of a number 3
triple of a number 6
triple of a number 9
triple of a number 12
triple of a number 15
triple of a number 18
NameError: name 'beginnertime' is not defined.

Since in the above code we have only written begin time instead of begin time we want to place beginner time.

o/p:-

[It will shows same as the above output, but inplace of Error it displays the following o/p].

total time taken 12.253324247024848413.

* With Multithreading :-

```
from threading import *
import time
def double(numbers):
    for i in numbers:
        time.sleep(1)
        print("double of that number", i*2)
def triple(numbers):
    for i in numbers:
        time.sleep(1)
        print("triple of a number", 3*i)
```

```
numbers = [1,2,3,4,5,6]
beginnertime = time.time()
t1 = Thread(target = double, args =(numbers,))
t2 = Thread(target = triple, args = (numbers,))
t1.start()
t2.start()
t1.join()
t2.join()
print("total time taken", time.time() - beginnertime)
```

O/p:-

```
triple of a number double of that number 32
double of that number triple of a number 46
triple of a number double of that number 96
triple of a number double of that number 128
double of that number triple of a number 812
triple of a number double of that number 15
10
triple of a number double of that number 1812
total time taken 6.142756700515747
```