Practical No 5

Aim : Write A Program To Implement Different Type Of Threads In Python

A. Single Thread

 # Without Using Threading

```
class A:
    def run(self):
        for i in range(5):
            print("John Doe")

class B:
    def run(self):
        for i in range(5):
            print("Jane Doe")

obj1=A()
obj2=B()

print("Without Using Threading")
obj1.run()
obj2.run()
```

 # Using Threading

```
from time import sleep
from threading import Thread

class A(Thread):
    def run(self):
        for i in range(5):
            print("John Doe (Thread)")
        sleep(1)

class B(Thread):
    def run(self):
        for i in range(5):
            print("Jane Doe (Thread)")
        sleep(1)

objT1=A()
objT2=B()

print("Using Threading")
objT1.run()
objT2.run()
```

```
Without Using Threading
John Doe
John Doe
John Doe
John Doe
John Doe
Jane Doe
Jane Doe
Jane Doe
Jane Doe
Jane Doe
Using Threading
John Doe (Thread)
John Doe (Thread)
John Doe (Thread)
John Doe (Thread)
John Doe (Thread)
Jane Doe (Thread)
Jane Doe (Thread)
Jane Doe (Thread)
Jane Doe (Thread)
Jane Doe (Thread)
```

B. Multitasking Thread

```
# Mutitasking Thread
from time import sleep
from threading import Thread

class A(Thread):
    def run(self):
        for i in range(5):
            print("John Doe ")
            sleep(1)

class B(Thread):
    def run(self):
        for i in range(5):
            print("Jane doe ")
            sleep(1)

t1=A()
t2=B()
t1.start()
t2.start()
t1.join()
t2.join()
print("I Am Main Thread")
```

```
John Doe
Jane doe
John Doe
Jane doe
John Doe
Jane doe
John Doe
Jane doe
John Doe
Jane doe
I Am Main Thread
```

C. Daemon Thread

```
from threading import *
from time import sleep

def thread1():
    for i in range(5):
        print("This Is Non-Daemon Thread")
        sleep(2)

T = Thread(target = thread1)
T.start()
sleep(5)
print("Main Thread Execution")
```

```
This Is Non-Daemon Thread
This Is Non-Daemon Thread
This Is Non-Daemon Thread
Main Thread Execution
This Is Non-Daemon Thread
This Is Non-Daemon Thread
```