Lab16. Concurrent Programming in Python

In [ ]:

```
VIVIYAN RICHARDS W           205229133
```

1. Create a global variable, rand_number = 0. Create a function generate() that will generate a random integer from 1 to 100 and update the global variable, rand_number. Create another function display() that will display the generated random number which is available in the global variable, rand_number. Create two threads each one for generate() and display() functions. Start threads and observe each thread performing their tasks

In [39]:

```
rand_num = 0
import random as rm

class Thread:
    def _init_(self,rand_num):
        self.num = rand_num

    def generate(self,rand_num):
        self.index = rm.randint(1,100)

    def display(self):
        print(self.index)
```

In [40]:

```
a = Thread()
a.generate(rand_num)
```

In [41]:

```
a.display()
```

```
57
```

2. Create a class SleepingThread which will sleep for a random period of time. It will print a message “Thread (number) sleeps (time) seconds" Start 5 SleepingThread classes and observe the message.

In [89]:

```
rand_num = 0
import random as rm
class SleepingThread:
    def _init_(Self,rand_num):
        self.num = rand_num

    def count(self):
        return self.count

    def display (self,rand_num):
        #self.count = self.count + 1
        self.r=rm.randint(0,1000)
        print("Thread",self.count,"sleeps",self.r,"seconds")
```

In [90]:

```
s = SleepingThread()
s.display(rand_num)
```

```
Thread <bound method SleepingThread.count of <__main__.SleepingThread object at 0x000002A
6DAAA3F70>> sleeps 819 seconds
```