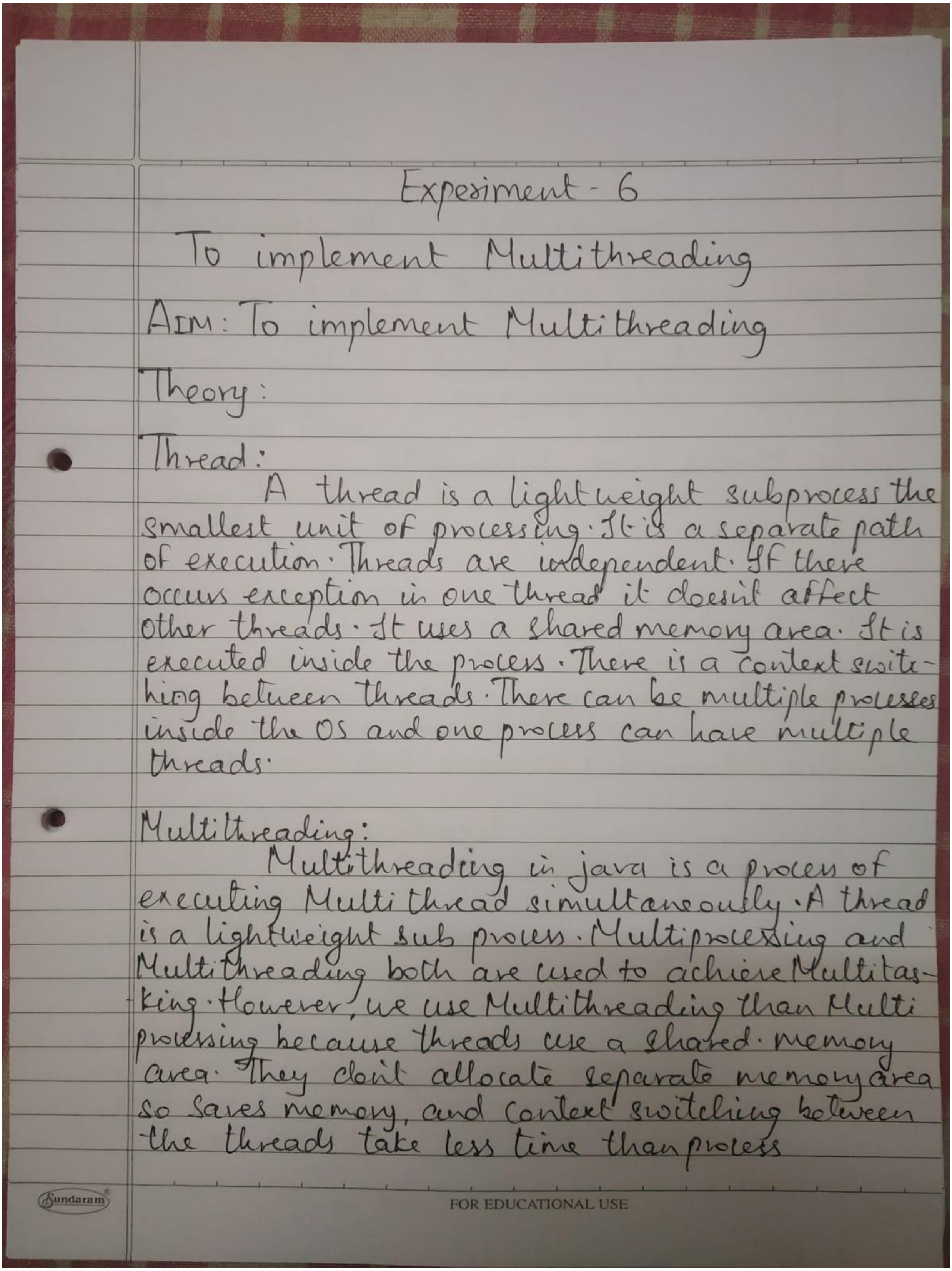
Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

COMPUTER ENGINEERING

OOPM ODD SEM 2021-22/EXPERIMENT 6

NAME:- GAURAV AMARNANI (D7A, 67)



Experiment - 6

To implement Multithreading

AIM: To implement Multithreading

Theory:

Thread:

A thread is a lightweight subprocess the smallest unit of processing. It is a separate path of execution. Threads are independent. If there occurs exception in one thread it doesn't affect other threads. It uses a shared memory area. It is executed inside the process. There is a context switching between threads. There can be multiple processes inside the OS and one process can have multiple threads.

Multithreading:

Multithreading in java is a process of executing Multi thread simultaneously. A thread is a lightweight sub process. Multiprocessing and Multithreading both are used to achieve Multitasking. However, we use Multithreading than Multi processing because threads use a shared memory area. They don't allocate separate memory area so saves memory, and context switching between the threads take less time than process

Sundaram

FOR EDUCATIONAL USE

Advantages of Multithreading

- It doesn't block the user because threads are independent and you can perform Multiple operation at the same time
- You can perform many operations together. so it saves time.
- Threads are independent. so it doesn't affect other threads if an exception occurs in a single thread

Java thread class

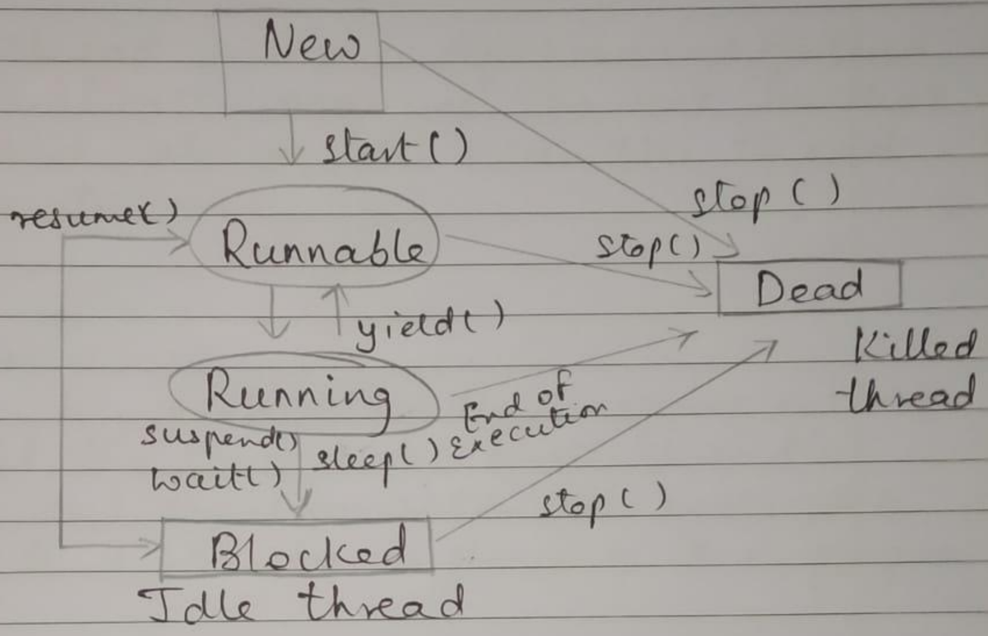
Java provides Thread class to achieve thread programming thread class provides constructors and methods to create and perform operations on a thread. Thread class extends object class and implement Runnable interface.

Life cycle of a Thread

In java, a thread always exists in any one of the following states. These states are:

- New
- Actie
- Blocked / waiting
- Timed Waiting
- Terminated

New
Thread



State Transition Diagram of Threads

Synchronization

Synchronization in java is the capability to control the access of Multiple threads to any shared resource. It is better option where we want to allow only one thread to access shared resource.

Java Synchronized Method

If you declare any method as synchronized, it is Known as synchronized Method. Synchronized method is used to look on object for any shared resources.

When a thread invokes a synchronized method, it automatically acquires the lock for that object and releases it when the thread completes its task.

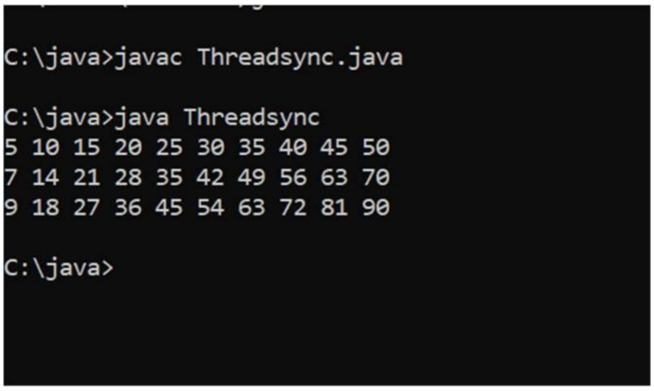
Conclusion:-
By performing this experiment we understood the concept of Multi threading. threads and synchronization life cycle of a thread. And How to implement it in program

Program:

```
class Table {
synchronized void printTable(int n) {
for(int i=1;i<=10;i++){
System.out.print(n*i+" ");
try{
Thread.sleep(400);
}catch(Exception e){System.out.println(e);}
}
System.out.println("");
}
}
class MyThread1 extends Thread{
Table t;
MyThread1(Table t){
this.t=t;
}
public void run(){
t.printTable(5);
}
}
class MyThread2 extends Thread{
Table t;
MyThread2(Table t){
this.t=t;
}
public void run(){
t.printTable(7);
}
}
class MyThread3 extends Thread{
Table t;
MyThread3(Table t){
this.t=t;
}
public void run(){
t.printTable(9);
}
}
public class Threadsync{
```

```
public static void main(String args[]){
Table obj = new Table();
MyThread1 t1=new MyThread1(obj);
MyThread2 t2=new MyThread2(obj);
MyThread3 t3=new MyThread3(obj);
t1.start();
t2.start();
t3.start();
}
}
```

Output-



```
C:\java>javac Threadsync.java

C:\java>java Threadsync
5 10 15 20 25 30 35 40 45 50
7 14 21 28 35 42 49 56 63 70
9 18 27 36 45 54 63 72 81 90

C:\java>
```