Date:
□□-□□-□□

Parallel Distributed computing
Assignment #1

1) Dispatcher code Explanation

Muhammad Iftikhar
P180054 6-A

```
final static int max-threads-At-A-time = 10;
Static int urrent Running Thread Count = 0;
static object dispatcher lock = new object();
Public static void main( String args[]) {
    For (Int i = 0, i < 100; i++) {
        Synchronized ( dispatch lock ) {
            urrent Running Thread - count ++;
        }
        final int this. Thread count += 1;
        new Thread(new Runnable() {
        @ override
        Public void run() {
            System.out.println ("Thread" + This. Threadcount
                                + " staying ");
            Try { thread.sleep( 5000); }
            catch ( Interrupted Exception e) {}}
        Synchronized (dispatchLock) {
            CurrentlyRunningThread count -- ;
            dispatcher .lock.notify () ; }
```

Date:
□□-□□-□□

```
}
}).start();
if (currentlyRunningThreadCount >= maxThreadsAtATime){
    synchronized (dispatcherLock) {
        try { dispatcher.wait(); }
        catch (InterruptedException e) { }
    }
}
}
}
```

100 threads are spawning 100 threads but limits it to run 10 threads only, The loop uses a counter currentlyRunning the count to check the Number of threads Running. It waits by calling wait() on lock object and as each thread finishes it's working, the counter is decremented and calls notify on the lock object. which enable dispatcher to make move threads.