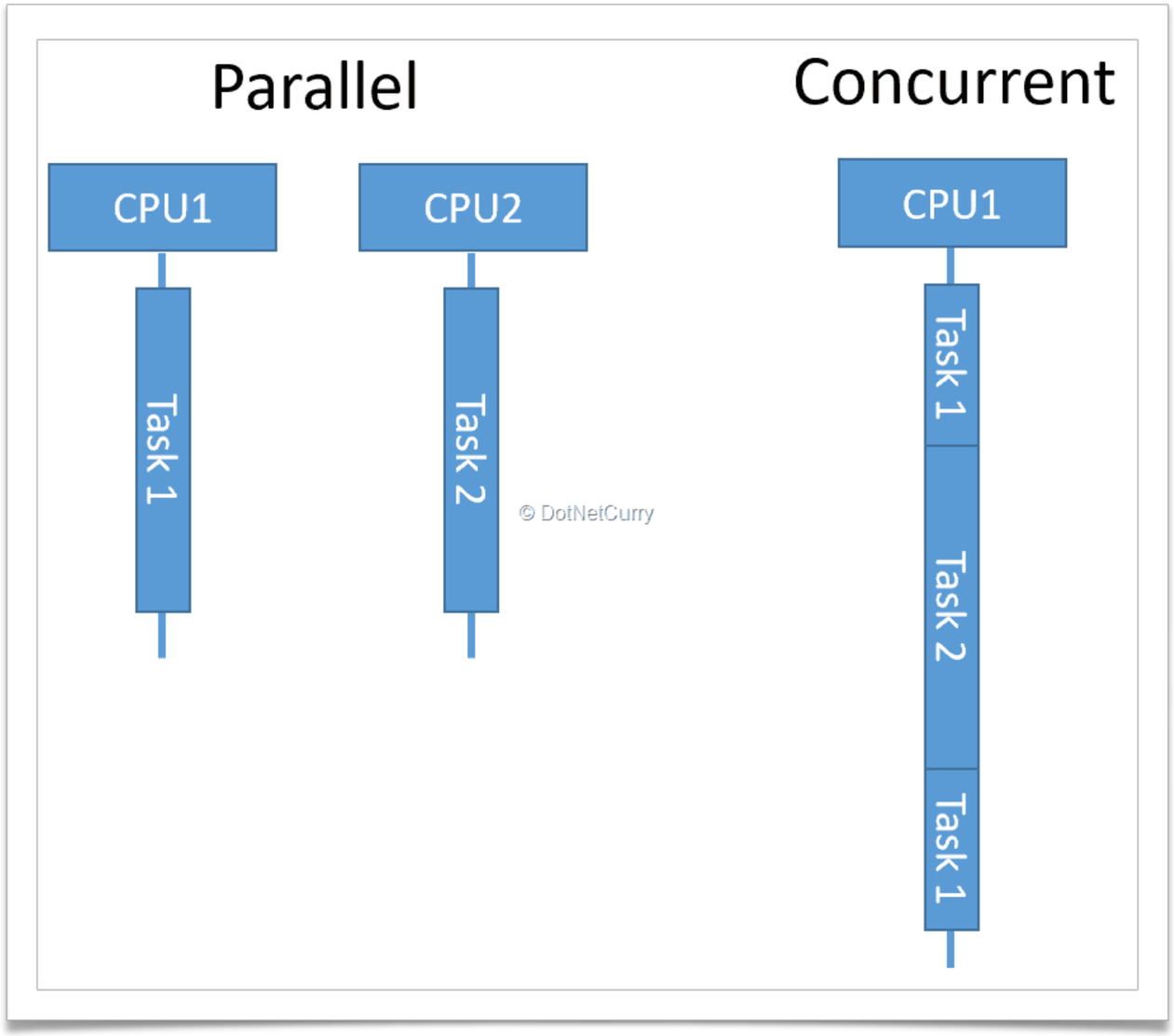
برنامه های موازی

مدرس: اسماعیل صادقی

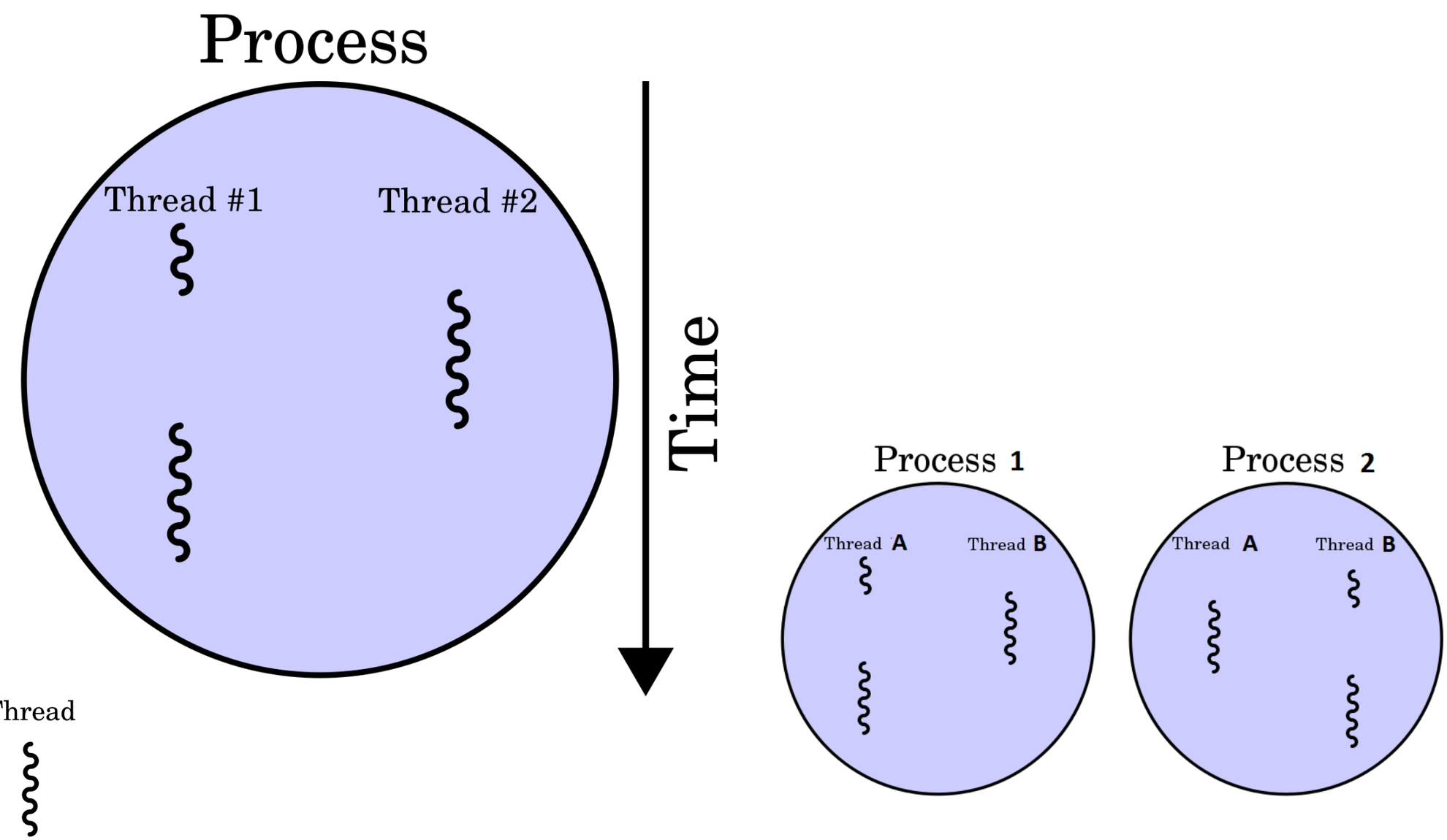
JavaTarFoundation

JavaTar

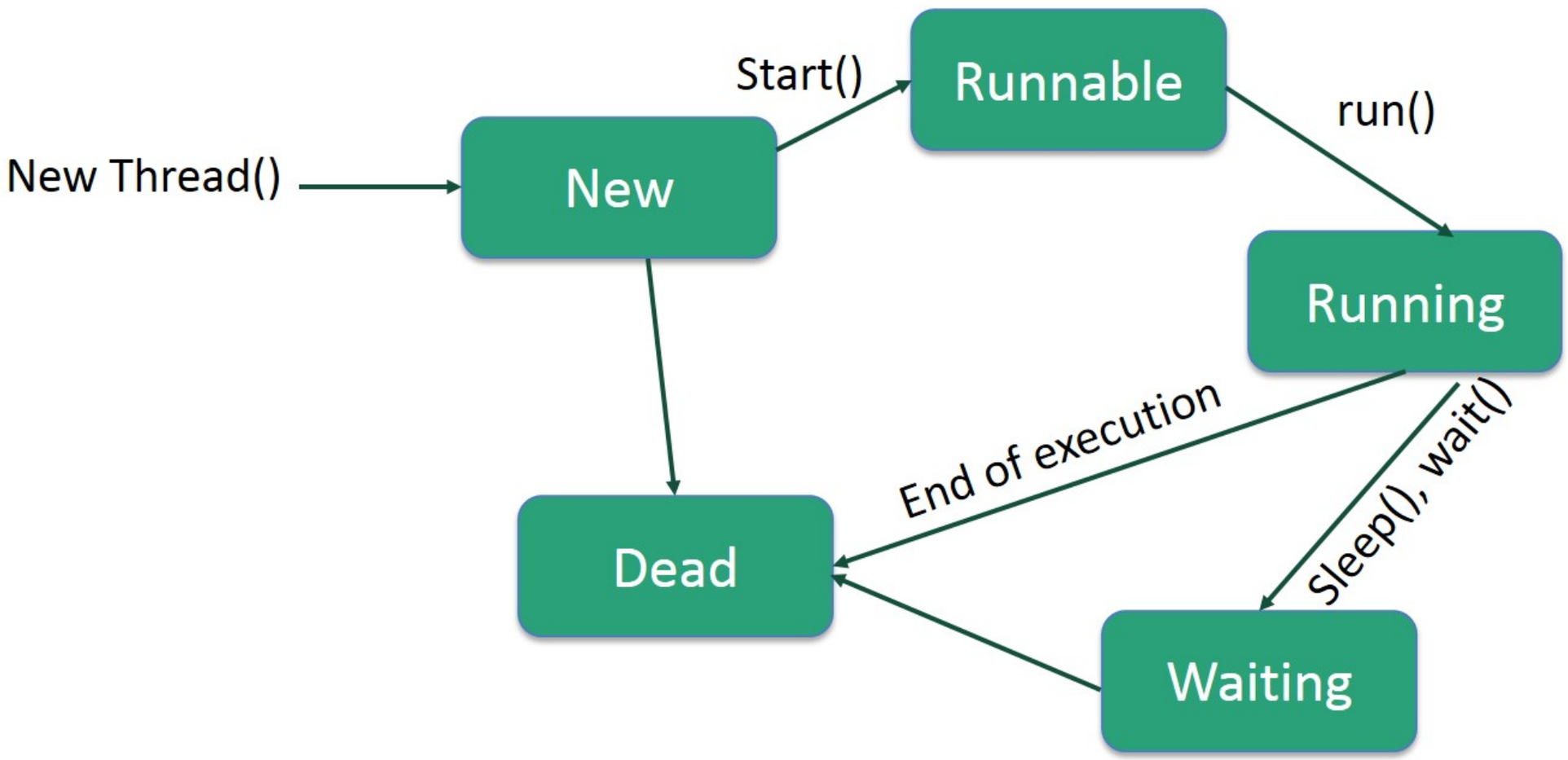
EsmaelSadeghijob@gmail.com



Thread

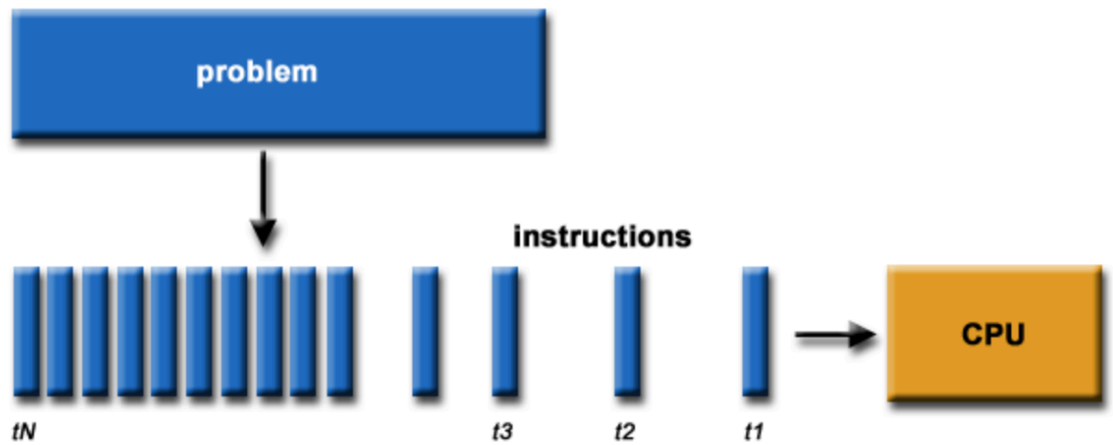


Thread States and Life Cycle

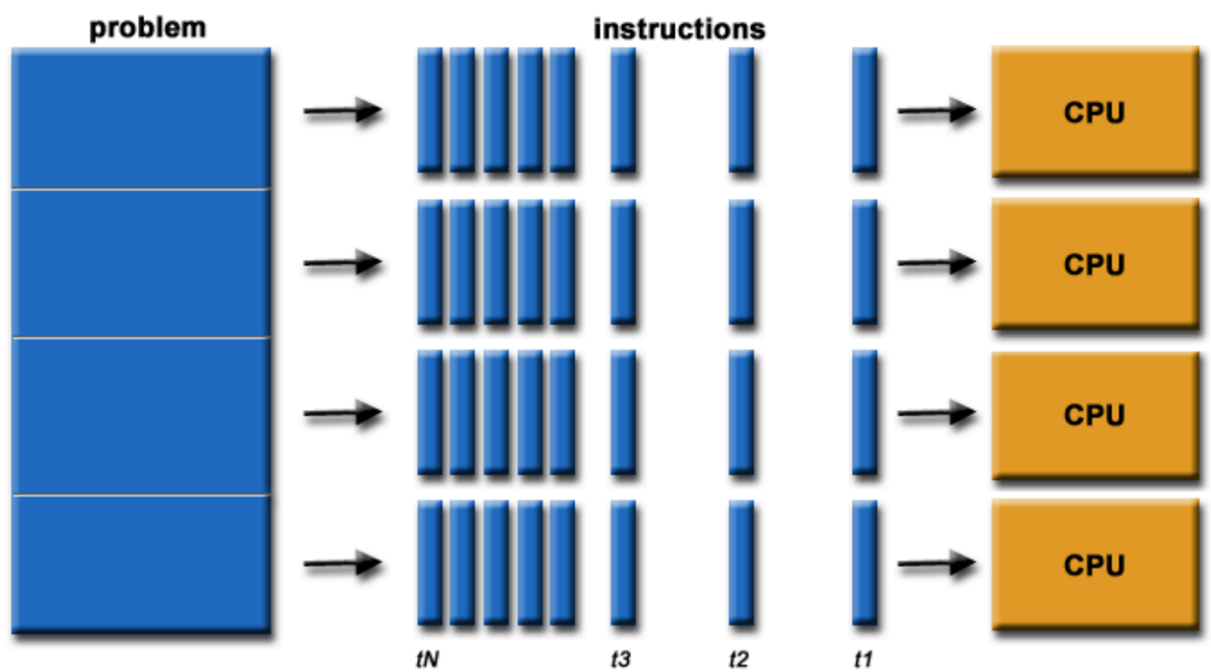


Between concurrency and parallelism?

Concurrency: If two or more problems are solved by a single processor.



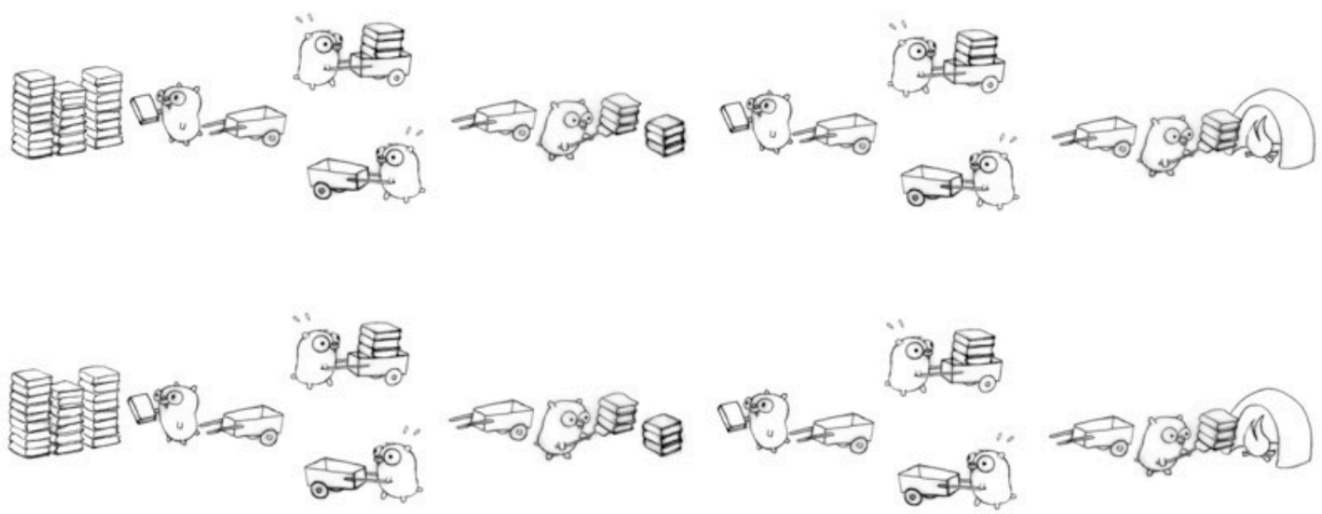
Parallelism: If one problem is solved by multiple processors.



Between concurrency and parallelism?

Task: Let's burn a pile of obsolete language manuals! One at a time!

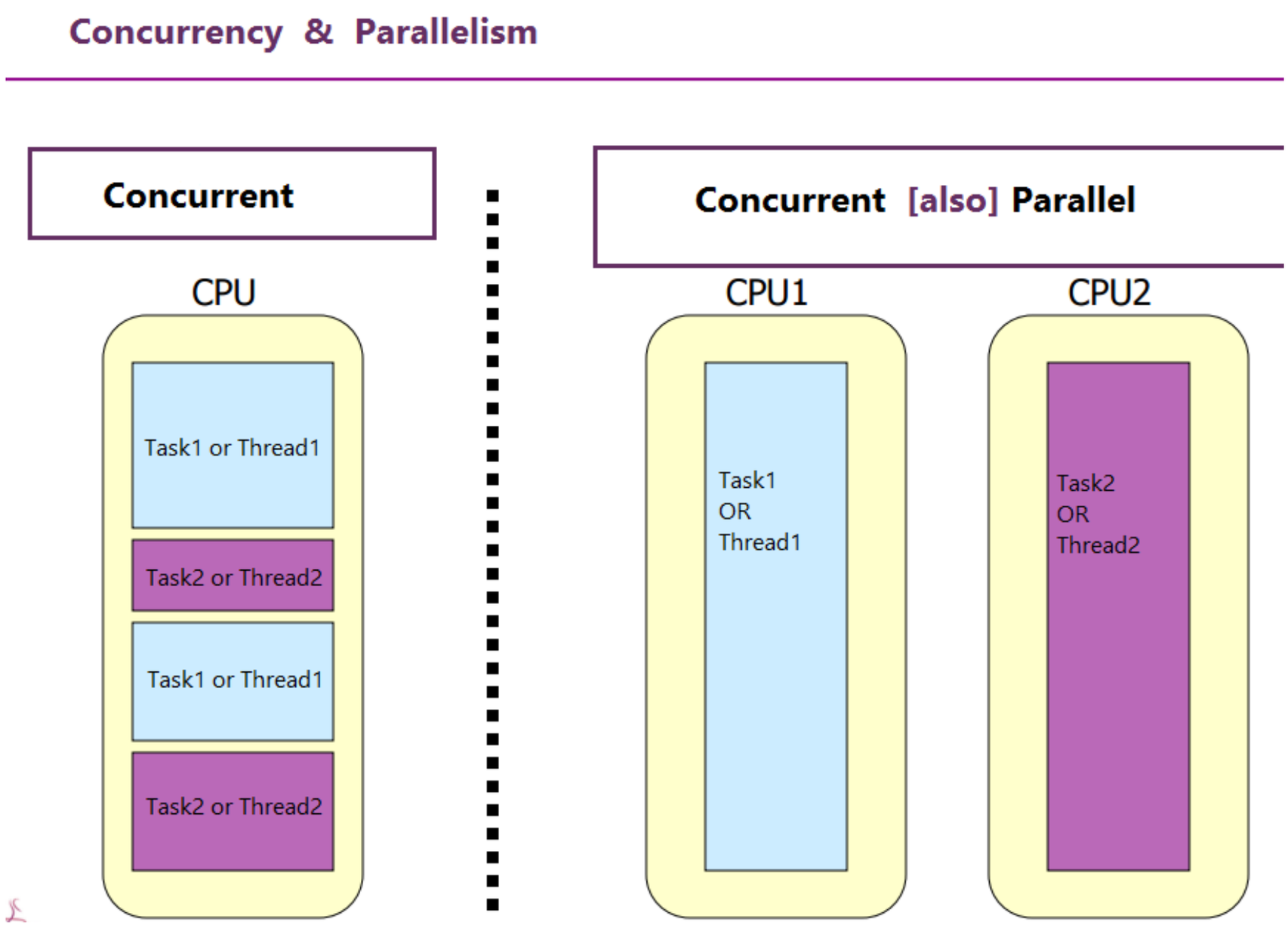
Concurrency: There are many concurrently decompositions of the task! One example:



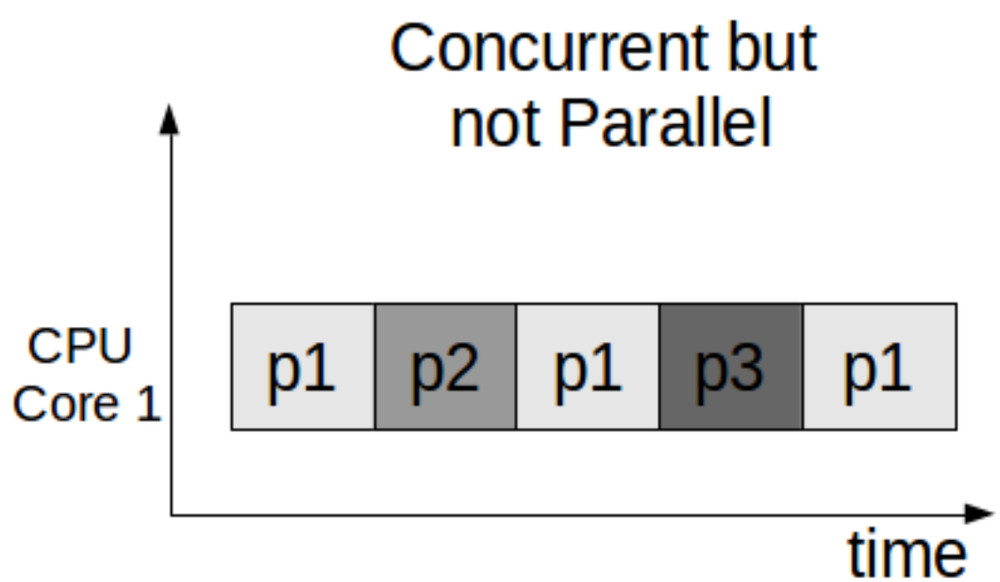
Parallelism: The previous configuration occurs in parallel if there are at least 2 gophers working at the same time or not.

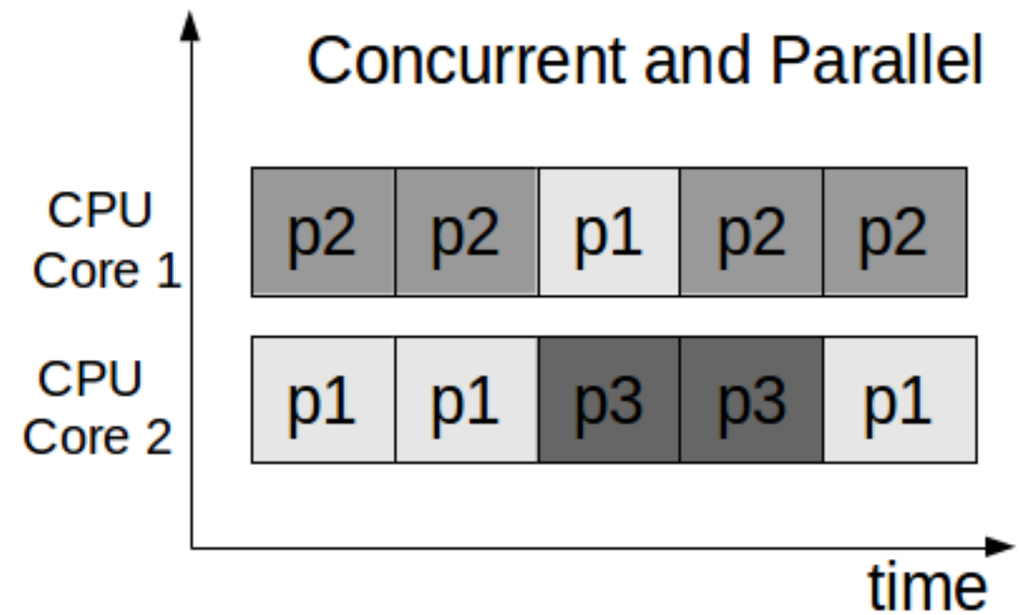
JavaTar
EsmaelSadeghi2ob@gmail.com

Between concurrency and parallelism?

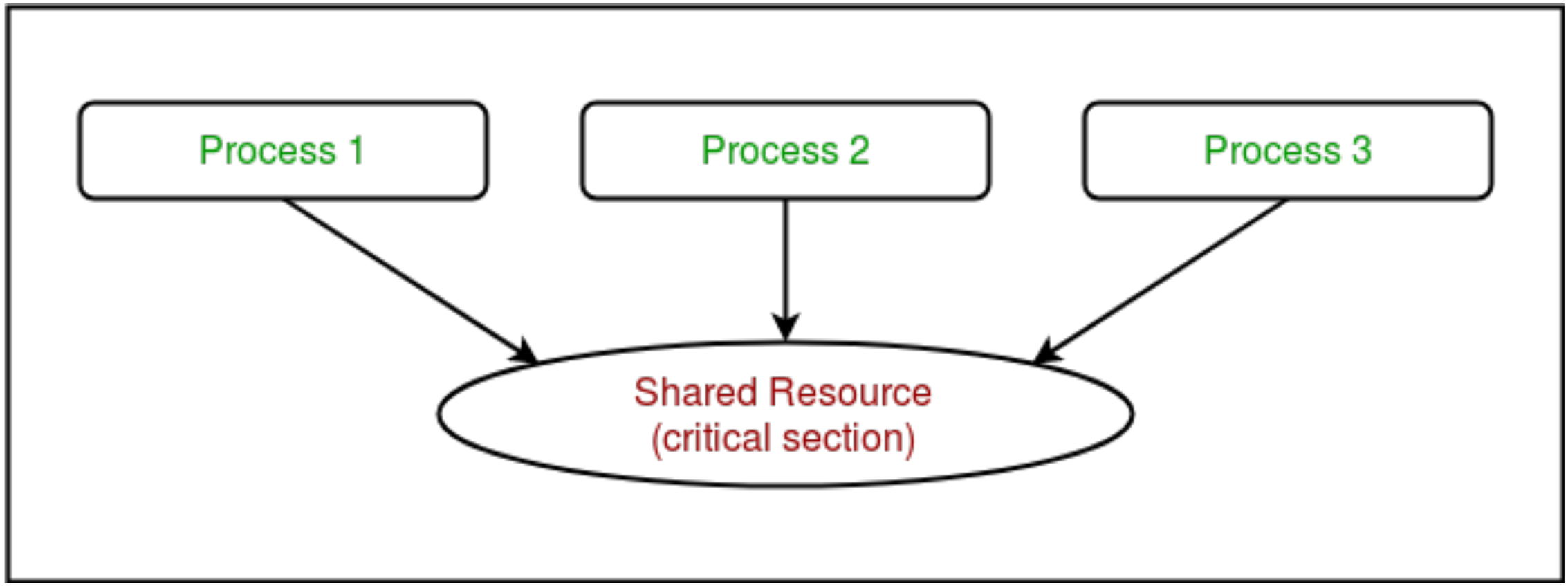


Between concurrency and parallelism?





Critical Section



synchronize

```java
public class UserMonitor {
 private int numUsers = 0;
 private volatile boolean occupied = false;



 public synchronized void addUser() {
  numUsers++;
  occupied = true;

 }

 public synchronized void deleteUser() {
  numUsers--;
  if (numUsers <= 0) {
   occupied = false;
  }
 }
}
```

synchronize

CounterThread.java

```java
public class CounterThread implements Runnable {

    private int count;

    public int getCount() {
        return count;
    }

    public void setCount(int count) {
        this.count = count;
    }

    @Override
    public void run() {
        Random rand = new Random();
        try {
            Thread.sleep(rand.nextInt(1000));
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        count++;
    }

    public static void main(String[] args) throws InterruptedException {
        CounterThread ct = new CounterThread();

        List<Thread> threads = new ArrayList<>();
        for (int i = 0; i < 100; i++) {
            Thread t = new Thread(ct);
            threads.add(t);
            t.start();
        }
        // wait for every thread to finish
        for (Thread t : threads) {
            t.join();
        }

```

Problems | Javadoc | Declaration | Console

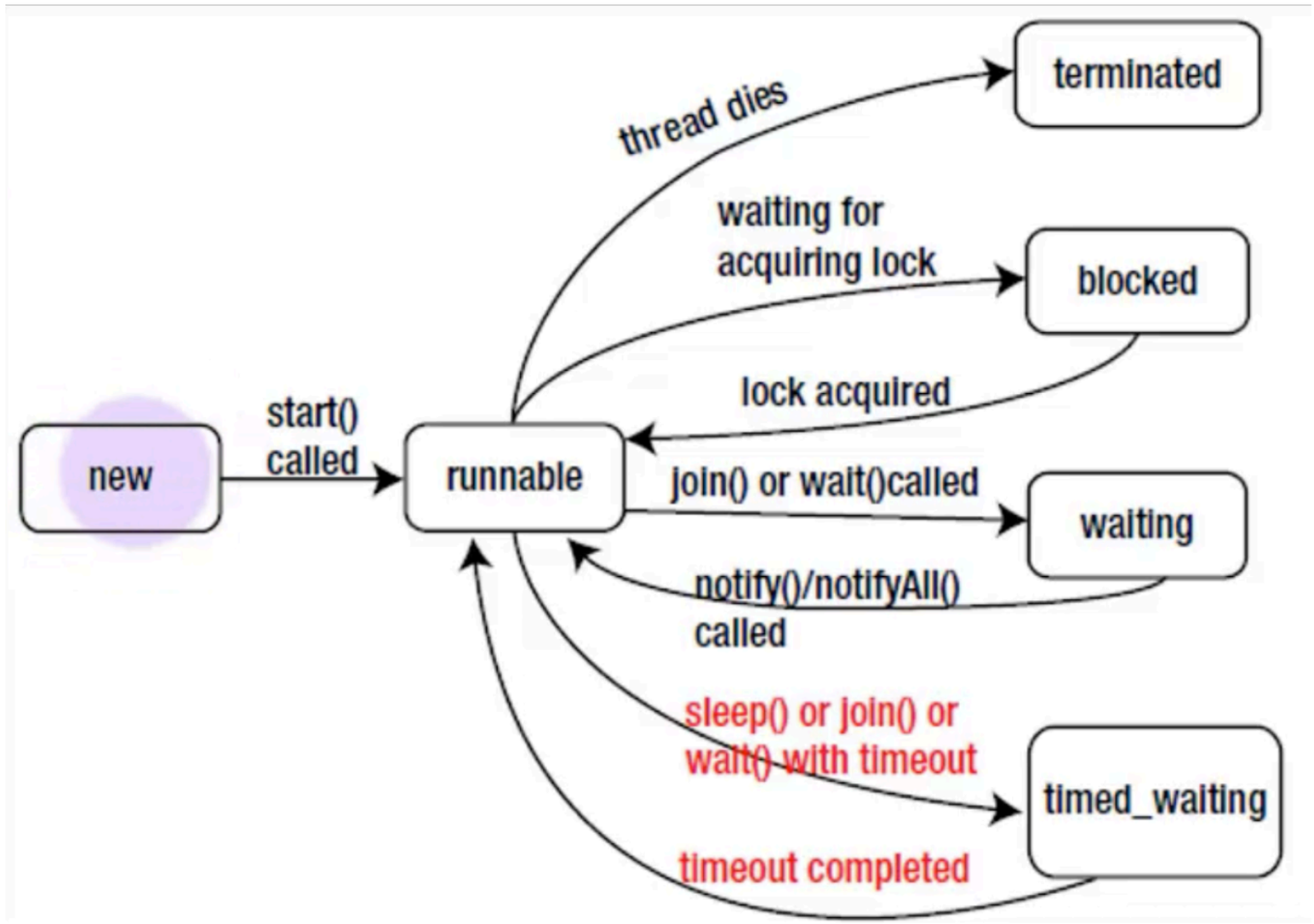
```
<terminated> CounterThread [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.jdk/Contents/Home/bin/java (08-Jul-2019, 1
Final Count = 91
```

Wait / notify

```
void waitForCompletion() {
  synchronized (lock) {
    if (!completed) {
        lock.wait();
    }
  }
}
```

```
void setCompleted() {
  synchronized (lock) {
    completed = true;
    lock.notifyAll();
  }
}
```

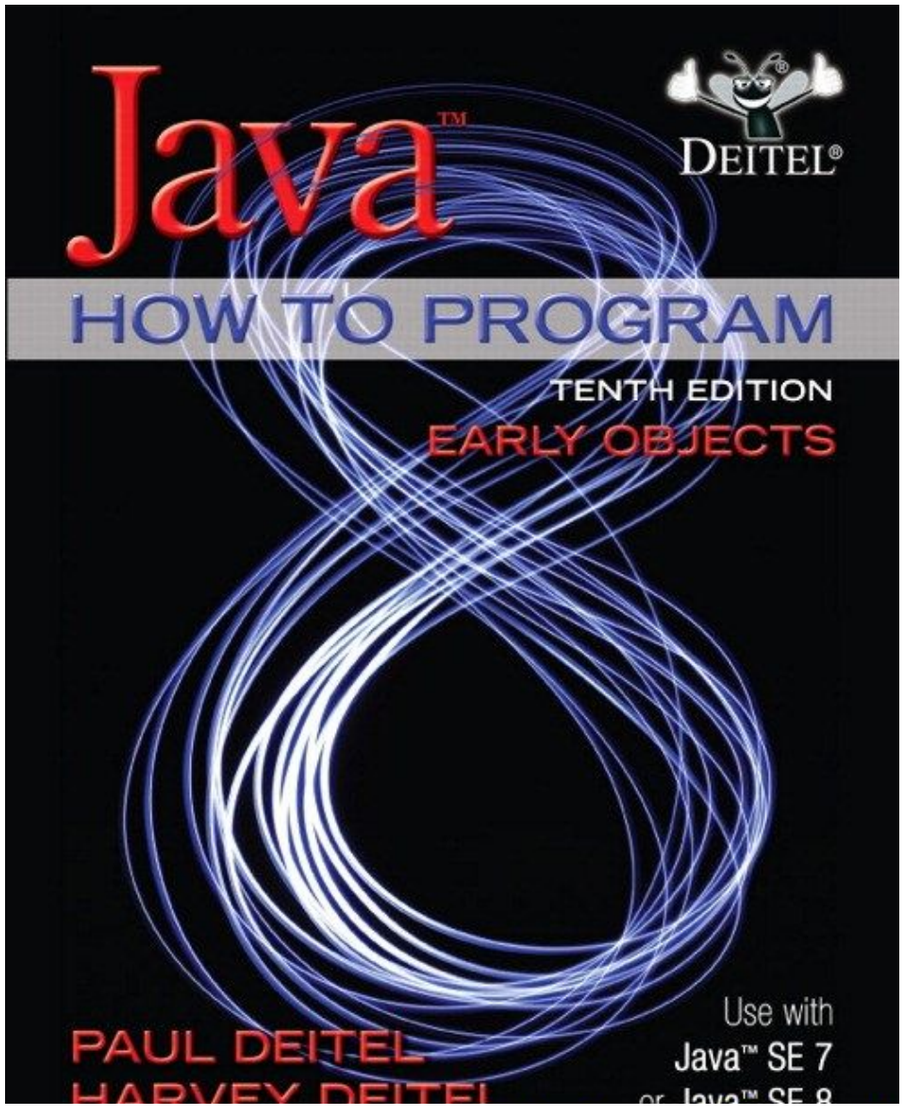
State Thread



Thread Safe

Non Thread safe in concurrency	Thread safe in concurrency
	Immutable (String , Integer) no setter
ArrayList	Vector
HashMap	concurrentHashmap
StringBuilder	StringBuffer

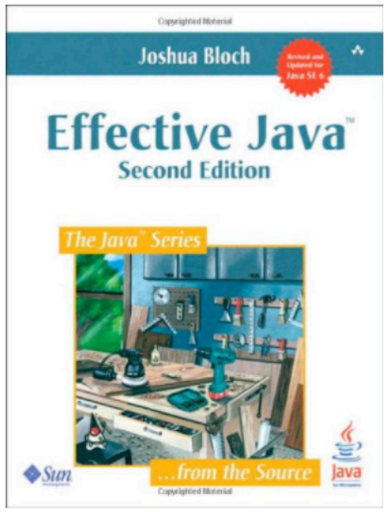
Book

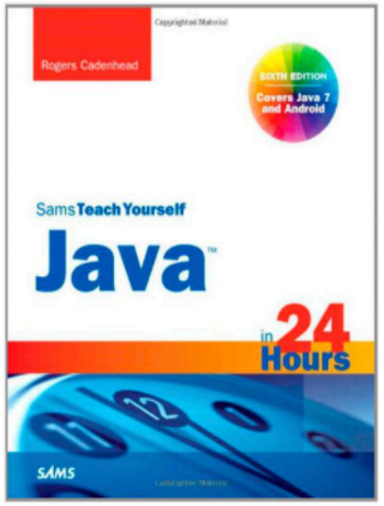


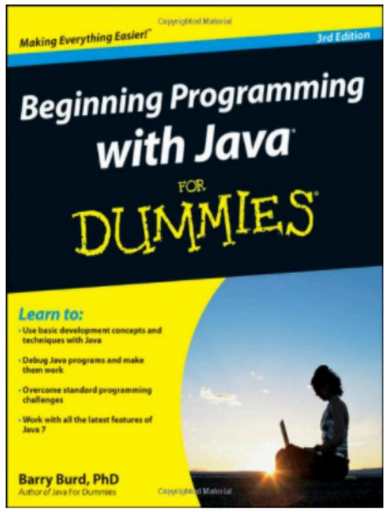
JavaTarFoundation

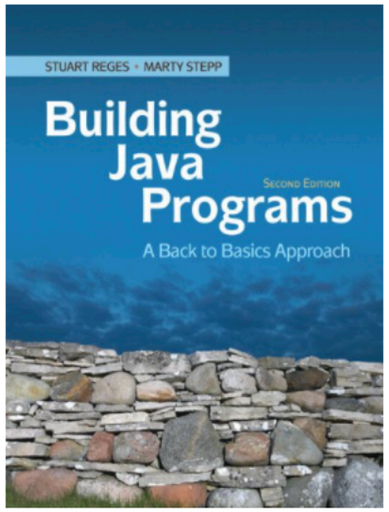
JavaTar
EsmaelSadeghi7ob@gmail.com

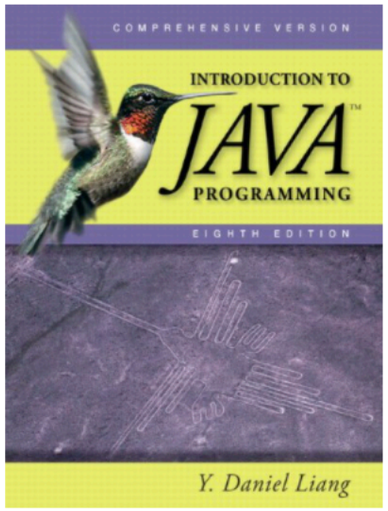
Useful Books on Java Concurrency











JavaTarFoundation

JavaTar

EsmaelSadeghijob@gmail.com