# Operating Systems (coe628)
# Lab 8

### Due the Week of March 18, 2024

## Description

Implement a basic Semaphore in Java. In order to "block" a Thread in Java, use the "waitO" method (inherited from Object). To wake up something, use the "notifyO" method. For example, the down method may block if the semaphore value is zero. In this case, use "waitO". The "up()" method increments the Semaphore value and never blocks. However, if the value was zero, there may be something waiting and "notify()" should be used.

### The Semaphore API

- Download the source files for this lab on D2L.
- You need to implement a semaphore (invented by Edsger Dijkstra), more information can be find at https://en.wikipedia.org/wiki/Semaphore_(programming)

### What you need to implement

Create a Netbeans project that implements the ProducerComsumer class that uses semaphores. The source files also include the ProducerComsumer class. Once you implement the Semaphore class, the ProducerComsumer project should work correctly!. In order to do that, you need to:

- Create a semaphore class as shown below
- Create a method *void up()* in the class, such that it increments the number of available resources. This method never blocks. It may wakeup a Thread waiting for the Semaphore. Dijkstra called this the V operation. Many also call it the signal or release methods.
- Create a method *void down()*, such that if a resource is requested, if no resources are available, the calling Thread blocks until a resource controlled by the Semaphore becomes available. Dijkstra called this the P operation. Many also call it the wait or the acquire methods.
- Ensure that your code is compiled and running.

```
/* Create a semaphore */
public class Semaphore {
private int value;

public Semaphore (int value) {
   this.value = value; }

public void up() {}
public void down() {}
}
```

## Submit your lab

On a departmental lab computer, do the following

a. Zip the submission folder:
zip -r coe628_lab8.zip coe628_lab8
b. Submit the folder:
submit coe628 lab8 coe628_lab8.zip