# Lab CC 6: Race Conditions

## Objectives

In this lab, you will create a race condition with multiple threads accessing a shared resource. In part two, you will use the synchronized keyword to manage the shared resource

## Part One: Race Condition

### Step 1: Create the Counter class

1. The counter class is passed a reference to a counter that it will increment then decrement.
2. This counter will be shared by 10 counter threads and will initially be set to zero
3. Since each thread adds 1 then subtracts 1 from the counter, after all of the threads have run, the counter should be back to 0.

Text

Description automatically generated

1. The run() method for the class is defined as follows

Graphical user interface, text, application

Description automatically generated

### Step 2: Create the runner class

1. This class starts 10 counter threads running and passes to each a reference to a single Counter object.

Text

Description automatically generated

### Step 3: Run the code

1. Eventually the counter may be zero. Notice that according to the code, it should never be more than one but the interleaving of operations means that several threads are incrementing before decrementing.
2. This is non-deterministic since we can never predict what the value of the counter is at any specific point in execution of the code. In the example below the final result is -1
3. Run the code several times and notice that the results will vary.

Table

Description automatically generated

## Part Two – Synchronize

A picture containing text

Description automatically generatedThis part adds the synchronized block to ensure that only one thread at a time can access the code that manipulates the counter block. This is in a class called SynchCounter

The only change to wrap the code in run() method in a synchronization block

Text

Description automatically generated with low confidence

And we change the runner class to use this new class.

Text

Description automatically generated

Text

Description automatically generatedRunning the code shows that now the counter objects work exactly as we would expect.