



# Concurrent Programming

## Pass Task 06: Mutex

### Overview

A Mutex is a special kind of Semaphore, one where there can be only 0 or 1 tokens.

### ***Pass Task 06 — Submission Details and Assessment Criteria***

You must submit the following files to Doubtfire:

- Mutex source code (Mutex.cs)
- Mutex Test class (MutexTest.cs)

### Instructions

Create a Mutex class in your *concurrent utilities library*. Implement the following features:

- Inherit from the Semaphore
- Override the Release(n) method — throw an exception if there is already a token
- The constructor should initialise the Mutex with 1 token.
- Ensure that all public features, including the class, have XML documentation.

Create a Mutex Test class that demonstrates the use and features of a Mutex. In this case the features you want to demonstrate are:

- If a Thread tries to release a token when there is one already present, an exception is thrown.
- You can acquire the Mutex when it is created.

**Note:** Make sure the output show if the Mutex is/isn't working correctly. Lots of fast output is bad!