



# Concurrent Programming

## Pass Task 02: Hello "Active Object" World

### Overview

As a first step, create the classic "Hello World" program with a concurrent twist — with a little more structure.

### *Pass Task 02 — Submission Details and Assessment Criteria*

You must submit the following files to Doubtfire:

- Hello World source code (HelloWorld.cs)
- Active object class (ActiveObject.cs)
- Message Writer object class (MessageWriter.cs)
- Screenshot of the Terminal showing output from program.

### Instructions

Active Objects have their own "lifeline" by encapsulating a thread.

- Create a class library project
  - Use a namespace that contains your name or student id. Eg: Acain.Concurrent.Utilis
  - Add an Active Object class:
    - Is abstract
    - Is given a name when created — assign it to the object's Thread.
    - Has a Thread that is created when it is constructed.
    - Provides the ability to **Start** the Active Object — which starts its Thread.
    - When started, it executes its protected abstract **Run** method.
    - Has XML documentation... that is useful!
- Create a program that uses the class library (in a different namespace/project)
  - Create a Message Writer Active Object class
    - Assign it a message when created (in the constructor).
    - Override Run and have it infinitely print its message to the Terminal.
  - Create a Hello World program that uses your Message Writer to output "Hello World" messages to the Terminal.