

# Advanced Programming Homework Assignment 10

Due Wednesday, April 9, at 3PM

The goal of this assignment is to implement a fixed-length concurrent queue abstraction on top of Haskell's `MVar` primitive.

1. Download and unpack the tar file containing the code template. Have a look at the files.
2. Try running the program (either using the supplied `Makefile` or by compiling and executing `TestHarness.hs` in whatever way you usually do). Notice that the test harness reports a failed test. This is because `FiniteChan.hs` currently contains an implementation of *unbounded* concurrent queues. Your job is to replace this implementation by a working one.
3. Haskell provides several different concurrency primitives (`MVars`, `STM`, higher-level synchronization abstractions in libraries, etc.), but for this assignment the only one you can use is `MVars`.
4. There are many ways to implement the required functionality using `MVars`. Do not worry about finding the most efficient one (or the most parallelizable one, etc.). Instead, aim to make your code not only correct by *obviously* correct. You'll probably find this is challenging enough!

## Submission instructions

- Add your name(s) at the top of your `FiniteChan.hs` file.
- Email just the file `FiniteChan.hs` to both `jschorr@seas.upenn.edu` and `bcpierce@cis.upenn.edu`. Make sure your file is named (exactly) `FiniteChan.hs`.