CSE P505, Spring 2006, Assignment 5
Due: FRIDAY 2 June 2006, 11:00PM

Last updated: May 19.
The 3 problems are independent; 2 and 3 are about CML. Only 2 requires you to write code.

1. Consider a class-based OO language supporting the declaration "`class C is D without m`." If `D` is a class with a method named `m`, then this declaration creates a class `C` that "is like `D` without a method `m`."

   (a) Decide if the meaning should be "instances of `C` *have* an `m`, but the type `C` does not reveal it" or "instances of `C` really have no reference to code for `m`." The declaration `class C is D without m` should be allowed for all `C`, `D`, and `m` and should not violate type-safety (assuming the language is otherwise type-safe). Justify your choice with technical arguments and examples.
   Hint: There is a right answer and a wrong answer.

   (b) Explain how your answer to the previous question would change (or not) if `D` was an interface or a class known to have all abstract methods.

   (c) Decide what if any subtyping relationship there should be between `C` and `D` given `class C is D without m`. Justify your choice with technical arguments and examples.

2. In `hw5b.ml`, reimplement the bank account example from lecture 9, which implements the interface in `hw5b.mli`. Still use Concurrent ML (the `Event` library), but use 4 channels (2 input and 2 output) instead of the `action` datatype. `get` operations should use different channels than `put` operations. Clients should not be able to distinguish your implementation from the one in lecture.
   Hint: You need to use `choose` and `wrap` in a straightforward way.

3. Document the function `mystery_f` in `hw5c.ml`. That is, in English sentences, describe its behavior, including

   - What it expects for arguments
   - What it does with messages from other threads

   Do *not* describe the exact procedure by which it implements its behavior. *Do* explain any behavior observable to callers or other threads. Do *not* just explain how the program `hw5c.ml` behaves (it does nothing and does not terminate). *Do* explain how `mystery_f` behaves in general.

4. **Extra Credit:** Make a second answer to problem 2 where the "get communication" use the same channel for input and output (rather than 2 distinct channels). This change is *wrong*. Write a Caml program that shows why this change is wrong. Your program should consistently have the wrong behavior in practice (not necessarily in theory); use calls to `Thread.yield`.

Turn in:

- Email your solutions to Ben. Include `hw5b.ml` as an attachment. Put your answers to problems 1 and 3 in a text, pdf, or Word document.
- Do the extra credit in a separate file `hw5ec.ml`.
- If you are using Seminal, please include your backup files.