Prepare for final

- **Time:** 5:00-6:50 Wed 23 DEC 2009.
- **Location:** Room 102

- Go over all hws
- Go over finals from the past 2 or 3 years.
- If you have more time, go over some of the questions from the dept. core exams.

Prepare for final

- Recent topics you need to know
  - Concurrency
  - Exception Handling
  - Generics
  - ADT
  - Inheritance
  - Dynamic dispatch

- To **know** means to understand the concept and writing simple code to explain the concept
Prepare for final

- Previous topics you need to know
  - Parameter passing: 4 modes
  - Scoping: static vs dynamic
  - Typing:
    - Inference
    - Subtyping
  - Lambda calculus: alpha and beta conversion
  - Functional languages
    - List related functions in ML and Scheme
      - Length(), append(), reverse()
    - Use recursion

Concurrency

- What is concurrency (Be sure to include in your answer the case in which there is only a single processor.)?

  Concurrency means the simultaneous execution of multiple portions of a program.
  - a logical perspective, not necessarily actual simultaneity

- Each portion is typically referred to as a “task” (as in Ada) or “thread” (as in Java).

- What does “Concurrency” mean in a single-processor machine?
  - Because tasks or threads might not actually execute simultaneously, as is the case on a single-processor machine, concurrency means that the relative order of events (e.g. instructions) in two concurrent tasks or threads is left unspecified.
- Write a simple example in Java or Ada of a program that exhibits concurrency.

- How to do synchronization to avoid bugs due to race conditions in Java?
  - **Object Synchronization**: Attaching "synchronized" to a non-static method \( G \) of a class \( C \) means that if, for a given object \( o \) of class \( C \), if \( o.G \) is running, then no other synchronized method of \( o \) can run.
    - `synchronized void G { .... }`
How to do synchronization to avoid bugs due to race conditions in Java?

- **Class-wide Synchronization**: Attaching "synchronized" to a static method of a class C means that, if f is running, then no other synchronized static method of class C can be running.

  ```java
class C {
  static synchronized void f() { ... }
  static void g() { // can not run if f is running
    ...
  }
}
```

- **Statement/Block synchronization**: Using the construct

  ```java
  synchronized(o) { ... }
  ```

  where o is an object, means that while the block ... is being executed, no other block that is also synchronized on the same object can be executing.

  ```java
class C {
  int[] a = new int[20];
  void f() {
    synchronized(a) { .... }
    ...
  }
}
```
- How to do synchronization to avoid bugs due to race conditions in Ada?

  - Using a task to control access to a variable:
  - Using a protected type, which ensures that only one procedure declared within the protected type can be running at any one time.