Course: 12A 3-11-16

- final exam: Wed Mar 16
  5:00 am

- PWB due by 1 day (only)

**Exercise:**

- Read Pair Counter

- Create DownCounter.java extend AbstractCounter from limit-1 down to 0
  - define click() appropriately
  - override reset() to start at limit-1
An interface in Java is a class-like structure that encapsulates only:

- constants
- abstract methods

i.e. all implementation details are determined.

Remarks:

- all interface methods are implicitly public, so usually don't bother to write public.
- same for keyword abstract.
- A descendant class of an interface is created using implements.

- Any such class is obligated to define all its inherited abstract methods.

**General form:**
```
// myInterface.java
interface myInterface {
    // constants
    // abstract methods
}
```
// myClass.java
class myClass implements myInterface

// fields & methods
// define all inherited abstract methods

// Notice:

- Interfaces can inherit from other interfaces (use extends)
- A class can extend only one parent class but it can implement any # of interfaces.
Ex. StudentEmployee

\[ \text{Person (interface)} \]

\[ \text{Student (interface)} \]
\[ \text{Employee (interface)} \]

\[ \text{StudentEmployee (class)} \]

Ex.

\[ \text{IType (interface)} \]

\[ \text{AType (class)} \]
\[ \text{BType (abstract class)} \]

\[ \text{CType (class)} \]
\[ \text{DType (class)} \]