- final exam: Wed. Mar 16
  7:30-10:30 PM
- will post review problems.
- TA evals
  Instructor evals (online now)
- 12B: still looking for more space.
  Join 12M worklists especially 12M-06.
Access Levels
widest to narrowest:

1) Public
   accessible to all Java classes.

2) Protected
   accessible to subclasses of this class, even if subclass is in a different package.

3) Package (default, package-private)
   accessible to other classes in this package.

4) Private
   accessible only within this class.
Abstract classes

An abstract method is just a function with no body. Use abstract keyword.

An abstract class is a class containing one or more abstract methods.
General form:

```java
abstract class MyClass {
    // fields & methods
    // at least one abstract method
    abstract type myMethod(...params...);
}
```

In C++ we call this a function prototype.

Remarks:

- **Abstract classes cannot be instantiated**, i.e. no constructor or new of.
- Any subclass of an abstract class (that is not itself abstract) is obligated to define all inherited abstract methods.
Exercise:

Write a class called DownCounter that starts at limit -1, then counts down to zero, decrementing by 1 on each click. This

Recall:

Dynamic method dispatch and Polymorphism: each object containing type information that the JRE can access and use to decide how to act upon an object.
class should Extend AbstractCounter.

- Read PainCounter example

Interfaces

an interface is a class-like structure that encapsulates only

- abstract methods
- constants (final member variables)

i.e. all implementation details are deferred.
- Interface methods are implicitly public, so `public` is typically not used.
- Same for `abstract`.
- Interface methods may not be static.

- A class that defines the methods in an interface, is said to implement that interface.

- Keywords: `interface implements`
General form

```java
// myInterface.java
interface myInterface {
    // constants
    // abstract methods
}

// myClass.java
class myClass implements myInterface{
    // definition of abstract
    // methods in myInterface
}
```
Note:

- A class can extend only one parent class.
- A class can implement many interfaces.

Ex. `StudentEmployee`

```
Person (interface)

Student (interface)  Employee (interface)

StudentEmployee (class)
```
Ex. `TypeTest.java`

```
interface IType {
    RType (abstract class)
    AType (class)
    BType (abstract class)
}

cType (class)  DType (class)
```