Inner classes & Lambda Expressions

Inner Class Access

- An inner class (ex: anon observer class) can access..

- Including:
  - Local variables & parameters;
  - Fields & methods of containing object.
  - Fields & methods of inner class
- How?
  - Inner class automatically..
    to containing object and needed local variables.

Inner Class and Final Local Variables

- Why can inner class access only final local variables?
  - ..
  - So parameters and local variables no longer exist. But, Java makes copy of needed local variables/parameters.
    - Called..
  - If variable not final, Java does not know which value to capture.
- Effectively Final (Java 8)
  - Detects if a variable..
  - Effectively final OK for capturing variable.

```
void foo(int x) {
    // Don't change x!
    // x = 42;
    myModel.addObserver(new DaObserver() {
        @Override
        public void dataChanged(int newVal) {
            System.out.println("x = " + x);
        }
    });
}
```

Lambda Expression (Java 8)

- Awkward to create anon classes for small interfaces
  - Lambda expressions can be used instead when..

Use an anon-inner class:

```
void foo(int x) {
	myModel.addObserver(new DaObserver() {
		@Override
		public void dataChanged(int newVal) {
			System.out.println(newVal);
		}
	});
}
```

Use a lambda expression:

```
void foo() {
	myModel.addObserver(
		);
}
```

Syntax:
arg -> statement