CS162 Week 3

Kyle Dewey

Overview

- Assignment 1 wrap-up
- Secure information flow tidbits
- Problem solving in Scala

Assignment 1

Secure Information Flow

Adding a Field for the Label

- Could add in the base class
- Could put in each derived class
- Functionally the same, but internally different

`pc` Stack

- Define an `object` named `pc`
- It internally has a mutable stack
- There are many ways to do this, but `scala.collection.mutable.Stack` is probably the easiest

test27.not

Any questions on secure information flow?

Problem Solving in Scala

Useful Scala Features

Call-by-Value

- Functions take in values which have already been evaluated, and act on those
- Typical of most languages

```
def foo(x: Int) = x + 5
def bar(x: Int) = 42
...
foo(12 + 3)
bar(8 * 2 * 7)
```

Call-by-Name

- Functions take **unevaluated expressions**
- Using these expressions triggers evaluation **for each use**

```
def foo(x: Int) = x + 5
def bar(x: Int) = 42
...
foo(12 + 3)
bar(8 * 2 * 7)
```

Significance

- In a pure language, this could affect program termination, but little else
- With mutable state, it can be used to define rich control flow operators
  - `while.scala`

Streams

- Streams are an infinite sequence of something
  - Positive integers, Fibonacci sequence, prime numbers, etc.
- Scala supports their use
  - `prime_numbers.scala`