

# 2013W1-lecture12

October 20, 2013

## Contents

<b>1</b>	<b>Question of the Day</b>	<b>2</b>
<b>2</b>	<b>Logistics</b>	<b>2</b>
2.1	Deadline for Bonus Point for handin of an OpenPGP key to the “key” assignment: Oct 11 . . . . .	2
2.2	Conceptual Assignment #2 solutions . . . . .	2
2.3	Programming Assignment #3 . . . . .	2
2.4	Midterm exam coming up! . . . . .	2
<b>3</b>	<b>Some Midterm Prep Time</b>	<b>2</b>
3.1	Find the appropriate part to change to eliminate <code>numSetC</code> from the core . . . . .	3
3.2	Make the change to the syntax! . . . . .	3
3.3	Alter the interpreter test cases to recognize the new core . . . . .	3
3.4	Alter the interpreter to work with <code>numC</code> rather than <code>numSetC</code> . . . . .	3
3.5	Alter the desugarer’s approach to <code>numS</code> . . . . .	3
3.6	Alter the desugarer’s approach to <code>numSetS</code> . . . . .	3
3.7	Determine whether anything else needs to change . . . . .	4
3.8	What should I try next? . . . . .	4
<b>4</b>	<b>Finishing that old lecture:</b>	<b>4</b>
<b>5</b>	<b>What have we learned today?</b>	<b>4</b>

## 1 Question of the Day

(With thanks to friend and CS grad student Dutch Meyer.)

I have a Java function like:

```
public static boolean isBizarre(float x) {  
    return x != x;  
}
```

What does `isBizarre` do?

SOLUTION

Well, it's bizarre. But among other things, returns false if `=x=` has the value `=NaN=` (i.e., "not a number"). Try: `=isBizarre(0.0f / 0.0f)=`.

## 2 Logistics

**2.1 Deadline for Bonus Point for handin of an OpenPGP key to the "key" assignment: Oct 11**

**2.2 Conceptual Assignment #2 solutions**

Should be posted by the time class starts!

**2.3 Programming Assignment #3**

Milestone due Friday! REMEMBER to include ALL team members' names in `README.txt`.

Full submission due 1 week from Friday!

**2.4 Midterm exam coming up!**

Pre-posted background is available from the notes section of the website.

## 3 Some Midterm Prep Time

We'll focus on the part that's worth the most marks!

### **3.1 Find the appropriate part to change to eliminate numSetC from the core**

We want to eliminate `numSetC` from the core and replace it by `numC`. What comments and/or data definitions represent the syntax should we change?

### **3.2 Make the change to the syntax!**

Go!

### **3.3 Alter the interpreter test cases to recognize the new core**

Add some `numC` test cases, however many you think are appropriate. If you don't do this first, you'll dive in before you *understand* the approach. (This one is fairly easy, as is our next big change, but the one after really benefits from some advance planning!)

You can assume someone already wrote all the other test cases! You'd adapt or delete their `numSetC` test cases.

### **3.4 Alter the interpreter to work with numC rather than numSetC**

Change the interpreter so that it works with the new syntax. Assume it already works with `combineC` and the other parts of the core syntax.

You should be able to make almost all of the changes by just using templates.. but try to think through what's happening at the same time. For example, have we lost any power?

### **3.5 Alter the desugarer's approach to numS**

This is the easier part of fixing desugaring.

START BY CHANGING TEST CASES! (Again, assume someone already wrote good test cases for the previous desugarer.)

### **3.6 Alter the desugarer's approach to numSetS**

This is the harder part of fixing desugaring! How will you put all those individual numbers together into a single non-deterministic number result?

As before: START BY CHANGING TEST CASES!

Test cases are tremendously valuable for checking that your code is correct. They're even *more* valuable for ensuring that you think through how your code should work before writing it.

### 3.7 Determine whether anything else needs to change

Go!

### 3.8 What should I try next?

- For the lazy/eager part: Pay attention today and work examples!
- For the min/max part: Maybe implement `min` and `max`?
- For the environments part: try changing it so we get *dynamic scope* rather than *static scope*.
- For the combineC part: now change the concrete syntax to allow ANY list of expressions inside a `(? ... ..)` rather than just a list of numbers.

## 4 Finishing that old lecture:

file:2013W1-lecture7.org

## 5 What have we learned today?

- From QotD: Yet again, that semantics of even simple operations can be surprising!
- From midterm prep: That we're on our way to being well-prepared for the midterm!