

Logistics, Exercises and Demos

Everyone should register for the course ‘SAV’ in

<http://moodle.epfl.ch>

whatAFunCourse

Please obtain “The Calculus of Computation” book

Bring your laptops whenever you can

If you do not know Scala, please take a look now

– can write ~Java, can write ~Haskell/Ocaml, see more at:

<http://scala-lang.org>

Example: [binary tree](#)

Membership in Binary Search Tree

```
sealed abstract class BST {  
  def contains(key: Int): Boolean = (this : BST) match {  
    case Node(left: BST,value: Int, _) if key < value => left.contains(key)  
    case Node(_,value: Int, right: BST) if key > value => right.contains(key)  
    case Node(_,value: Int, _) if key == value => true  
    case e : Empty => false  
  }  
}  
case class Empty extends BST  
case class Node(val left: BST, val value: Int, val right: BST) extends BST
```

Java

```
sealed abstract class BST {  
}  
class Empty extends BST {}  
class Node extends BST {  
    public final BST left, right;  
    public final int value;  
    Node(lft : BST, vl : Int, rght: BST) {  
        left = lft; value = vl; right = rght;  
    }  
    public boolean equals(BST other) { ... }  
    public String toString() { ... }  
}
```

and this is all without the traversal...

Other Functional Languages

Objective Caml:

```
type tree =  
  Leaf  
  | Node of tree * int * tree
```

Haskell

```
data Tree =  
  Leaf  
  | Node tree int tree
```

Rest of Today's Class

Bookmark this site, gives demo of many interesting related tools:

<http://rise4fun.com>

Check course page and moodle page regularly

<http://lara.epfl.ch/w/sav>

Attend all lectures, exercises, labs that you can

Next Eva leads exercises/labs using Spec#