CSE 341  
Section 3

HW1 Tips  
Tail Recursion  
Pattern-Matching

Agenda

- HW1 tips  
  - Avoid reimplementation of functions  
- Pattern-matching over expression trees  
- A little tail recursion  
  - What is and isn’t tail recursive?  
  - How can we make functions tail recursive?

Homework 1 Check-In

Things to keep in mind:

- Easy to miss things when learning a new language!  
- Test as you go & design your tests carefully  
- Read the spec before, during, and after each problem. It’s worth the extra time!  
  - Check for correctness and redundancy/style (and review the CSE 341 Style Guide before submitting!)  
- Functional programming emphasizes code reuse  
  - Avoid re-implementing functionality where possible  
  - Will become even more important later with higher-order functions

Key Concepts Review

- Custom datatypes  
  - all of (records), one of (variants)  
- Pattern matching  
  - Powerful way to break apart data  
- Tail recursion  
  - Space efficiency of loops with recursive functions

val-Pattern Matching

Remember our unit test?

```scala
(* Neat trick for creating hard-fail tests: *)
val true = ((4 div 4) = 1);
```

Just a pattern match!  
“Match the left hand side against the value ‘template’ true, binding any variables (there aren’t any)!”

Adventures in pattern matching

- Shape example  
- Function-pattern syntax if we get to it
**Pattern Matching**

- We can pattern match over **datatypes**
- Beware “non-exhaustive matching”
  - Pattern matching can avoid “empty list” exceptions!
- Most functions pattern match over a single argument
  - SML has special syntax for this common case!
  - Use is a matter of taste
- Let’s work through some examples!

---

**Tail Recursion**

What is it?

Briefly: if a function will immediately return after making a call, we can reuse the stack space of the current function.

---

**Tail Recursion**

Quickcheck!

- Discuss the problems with your breakout rooms!

---

**Tail Recursion**

- Was `length` tail recursive?
- Was `all_positive` tail recursive?
- Why tail recursion?