

# 3 – Types, pattern matching, and the ALU

Eric Mayhew & Jacob Errington

In this lesson, participants are introduced to two key concepts of functional programming: pattern matching and recursion. By reimplementing functions for the standard library, participants will focus on the syntax for defining functions in Haskell. This will include recursion and pattern matching, the only control flow mechanism in Haskell. Finally, this lesson will wrap up hardware by covering the Arithmetic and Logic Unit, as participants create a half adder on their breadboard.

## Overview

### Learning objectives

By the end of the lesson, students will be able to:

- Read Haskell type signatures, identifying: function inputs and outputs, type variables, class constraints.
- Write Haskell functions using pattern matching on inputs.
- Reimplement several functions from the standard library, including recursive functions.
- Demonstrate how circuitry can perform arithmetic by building a half-adder using logic gate ICs.

### Materials

To run this lesson, the following materials are necessary:

- The lesson slides, <https://www.computing-workshop.com/pdf/{3-presentation.pdf}>.
- One breadboard for each group and enough of the following electronic components: NPN transistors, 100  $\Omega$  resistors, 9 V batteries, 5 V voltage regulators, jumper wires (male to male), assorted LEDs, integrated circuits from the 7400-series (7408 (quad 2-input AND gate), 7486 (quad 2-input XOR gate)).
- Each participant needs a computer to work on, with access to the web.

### Instructional sequence

- Instructional sequence is outlined in the slides.

### Homework