Lab 3: Higher-order functions and a simple interpreter

In this lab, you'll get some practice with higher-order functions, and then use what you've learned so far to build an *interpreter* for a simplified set of Racket expressions,including basic name lookup using a hash table implementation of an environment.

Starter code

- lab3.rkt

Tip: you may want to read up on Racket's pattern-matching syntax. You're welcome to use `define/match` instead of `define` for all functions in this lab (and in the future as well).

Task 1: Practice with higher-order functions

Inside the starter code, implement the three higher-order functions described in the file.

Task 2: A simple interpreter

In computer science, an *interpreter* is a program that takes another program as input and executes or evaluates the contents of that program. Of course, what we mean by "execute" or "evaluate" depends on the semantics of the programming language!

In this task, you'll write an interpreter of a simple arithmetic language and literals, with the twist that we'll be using strict Racket parenthesized-expression syntax, but writing operators in infix position.

Task 3: Representing an environment

Recall from lecture that an *environment* is a mapping of identifier names to values. For example, to evaluate the Racket expression `(+ 3 x)`, we must first determine the value of `x` (or raise an error if it is unbound). In an interpreter, we say that the value of `x` must be "looked up"—the *environment* is the data structure where the lookup occurs. We'll use the Racket *hash table* data type to store an environment; this data type has a similar interface to a Python dictionary or Java HashMap, with the usual restriction that we won't be using any mutating functions.

Your task here is to extend your interpreter from Task 2 to one that uses an additional environment argument to evaluate identifiers. We'll later explore *how* this environment is actually created, but don't worry about that for now.

Computer Science
UNIVERSITY OF TORONTO

For course-related questions, please contact **david at cs.toronto.edu**.