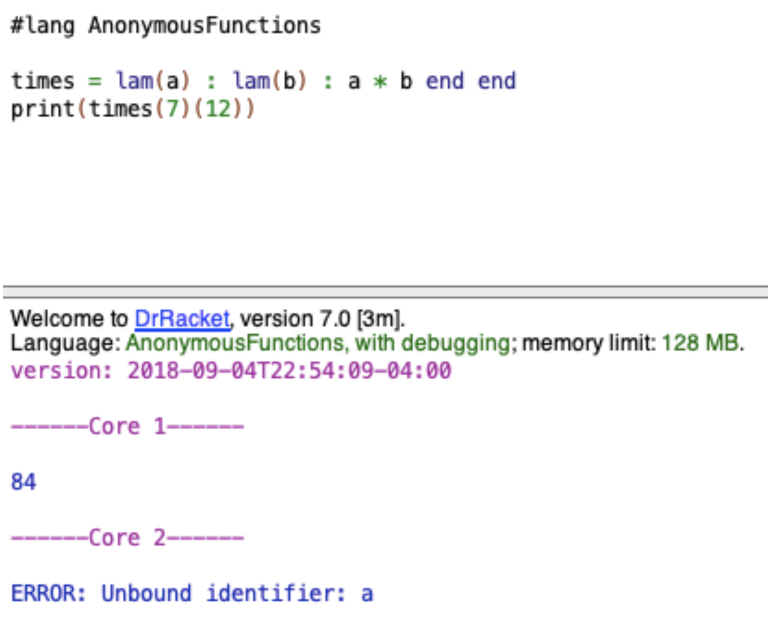
ENGF0002 (Design and Professional Skills)

Scenarios

This document focuses on the difference in the implementation of first class functions (i.e. lambdas) in the two prototypes of the programming language. It follows an observation-explanation-conclusion structure in that an observation is laid down, explained, and a theory is formulated it out of it.

Classifier-1 (1.rkt):



```
#lang AnonymousFunctions

times = lam(a) : lam(b) : a * b end end
print(times(7)(12))
```

```
Welcome to DrRacket, version 7.0 [3m].
Language: AnonymousFunctions, with debugging; memory limit: 128 MB.
version: 2018-09-04T22:54:09-04:00

------Core 1------

84

------Core 2------

ERROR: Unbound identifier: a
```

Observation-1: When a curried function is implemented in the language using lambdas (or first-class functions), then Core-1 returns a valid result. Core-2, however, fails to implement the currying.

Theory-1: A curried function is one which takes multiple arguments at a time and then translates it into sequence of functions, each taking one argument at a time.

As seen on the left, when such a function is implemented in this language, then Core-1 implements it correctly; takes one argument, a, and returns a function that takes another argument, b, and that function returns their product.

Core-2, however, is unable to implement the currying correctly due to scope issues as it returns an error- stating that *a* is not bound to the current scope- and fails to compile.

Partition after test: {Core 1}, {Core 2}